

```
In [ ]: import rl_test
        from utils import *
```

Reinforcement Learning Part 1: DQN

By Lawrence Liu and Tonmoy Monsoor

Some General Instructions

- As before, please keep the names of the layer consistent with what is requested in model.py. Otherwise the test functions will not work
- You will need to fill in the model.py, the DQN.py file, the buffer.py file, and the env_wrapper.py

DO NOT use Windows for this project, gymnasium does not support windows and installing it will be a pain.

Introduction to the Environment

We will be training a DQN agent to play the game of CarRacing. The agent will be trained to play the game using the pixels of the game as an input. The reward structure is as follows for each frame:

- -0.1 for each frame
- +1000/N where N is the number of tiles visited by the car in the episode

The overall goal of this game is to design an agent that is able to play the game with an average test score of above 600. In discrete mode the actions can take 5 actions,

- 0: Do Nothing
- 1: Turn Left
- 2: Turn Right
- 3: Accelerate
- 4: Brake

First let us visualize the game and understand the environment.

```
In [ ]: import gymnasium as gym
        import numpy as np
        env = gym.make('CarRacing-v2', continuous=False, render_mode='rgb_array')
        env.seed(42)
```

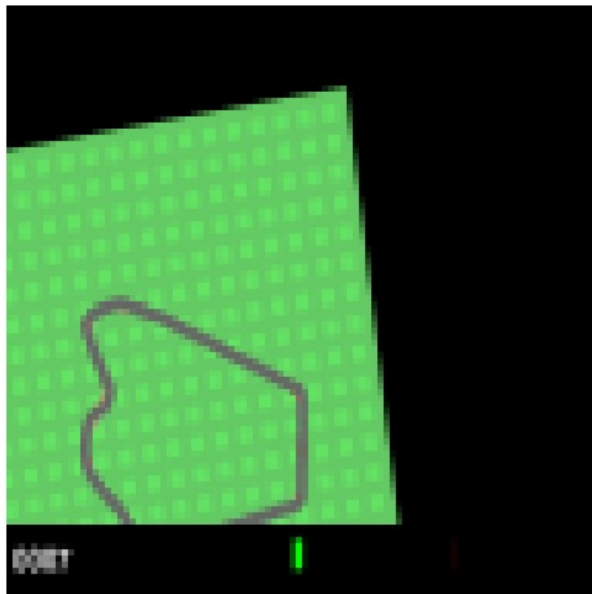
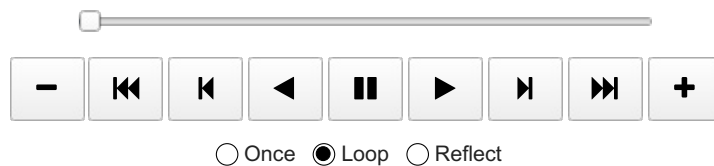
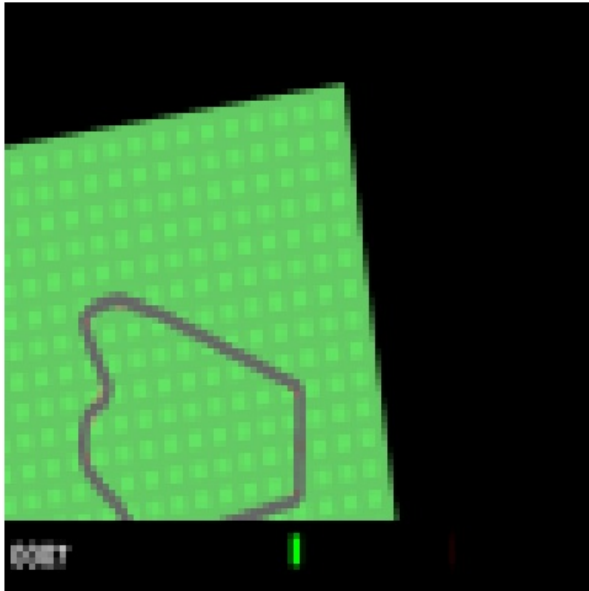
```
In [ ]: from IPython.display import HTML

        frames = []
        s, _ = env.reset()

        while True:
            a = env.action_space.sample()
            s, r, terminated, truncated, _ = env.step(a)
            frames.append(s)
            if terminated or truncated:
                break

        anim = animate(frames)
        HTML(anim.to_jshtml())
```

Out[]:



So a couple things we can note:

- at the beginning of the game, we have 50 frames of the game slowly zooming into the car, we should ignore this period, ie no-op during this period.
- there is a black bar at the bottom of the screen, we should crop this out of the observation.

In addition, another thing to note is that the current frame doesn't give much information about the velocity and acceleration of the car, and that the car does not move much for each frame.

Environment Wrapper (5 points)

As a result, you will need to complete `EnvWrapper` in `env_wrapper.py`. You can find more information in the docstring for the wrapper, however the main idea is that it is a wrapper to the environment that does the following:

- skips the first 50 frames of the game
- crops out the black bar and reshapes the observation to a 84x84 image, as well as turning the resulting image to grayscale
- performs the actions for `skip_frames` frames
- stacks the last `num_frames` frames together to give the agent some information about the velocity and acceleration of the car.

```
In [ ]: from env_wrapper import EnvWrapper

rl_test.test_wrapper(EnvWrapper)
```

Passed reset
Passed step

CNN Model (5 points)

Now we are ready to build the model. Our architecture of the CNN model is the one proposed by Mnih et al in "Human-level control through deep reinforcement learning". Specifically this consists of the following layers:

- A convolutional layer with 32 filters of size 8x8 with stride 4 and relu activation
- A convolutional layer with 64 filters of size 4x4 with stride 2 and relu activation
- A convolutional layer with 64 filters of size 3x3 with stride 1 and relu activation
- A fully connected layer with 512 units and relu activation
- A fully connected layer with the number of outputs of the environment

Please implement this model `Nature_Paper_Conv` in `model.py` as well as the helper `MLP` class.

```
In [ ]: import model
rl_test.test_model_DQN(model.Nature_Paper_Conv)
```

Passed

DQN (40 points)

Now we are ready to implement the DQN algorithm.



Replay Buffer (5 points)

First start by implementing the DQN replay buffer `ReplayBufferDQN` in `buffer.py`. This buffer will store the transitions of the agent and sample them for training.

```
In [ ]: from replay_buffer import ReplayBufferDQN
rl_test.test_DQN_replay_buffer(ReplayBufferDQN)
```

Passed

DQN (15 points)

Now implement the `_optimize_model` and `sample_action` functions in `DQN` in `DQN.py`. The `_optimize_model` function will sample a batch of transitions from the replay buffer and update the model. The `sample_action` function will sample an action from the model given the current state. Train the model over 200 episodes, validating every 50 episodes for 30 episodes, before testing the model for 50 episodes at the end.

```
In [ ]: import DQN
import utils
import torch

trainerDQN = DQN.DQN(EnvWrapper(env),
                      model.Nature_Paper_Conv,
                      lr = 0.00025,
                      gamma = 0.95,
                      buffer_size=20000,
                      batch_size=16,
                      loss_fn = "mse_loss",
                      use_wandb = False,
                      device = 'cuda',
                      seed = 42,
                      epsilon_scheduler = utils.exponential_decay(1, 700, 0.1),
                      save_path = utils.get_save_path("DQN", "./runs/"))

trainerDQN.train(200, 50, 30, 50, 50)
# trainerDQN.train(7, 2, 1, 1, 10)
```

saving to ./runs/DQN/run1

```

/content/drive/MyDrive/DQN_Project4/DQN.py:219: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
    states = torch.tensor(states).clone().detach().to(self.device)
/content/drive/MyDrive/DQN_Project4/DQN.py:220: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
    actions = torch.tensor(actions).clone().detach().to(self.device)
/content/drive/MyDrive/DQN_Project4/DQN.py:221: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).

    rewards = torch.tensor(rewards).clone().detach().to(self.device)
/content/drive/MyDrive/DQN_Project4/DQN.py:222: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
    next_states = torch.tensor(next_states).clone().detach().to(self.device)
/content/drive/MyDrive/DQN_Project4/DQN.py:223: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
    dones = torch.tensor(dones).clone().detach().float().to(self.device)
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/conv.py:456: UserWarning: Plan failed with a cudnnExcep
tion: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor Failed cudnn_status: CUDNN_STATUS_NOT_
SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
    return F.conv2d(input, weight, bias, self.stride,
/usr/local/lib/python3.10/dist-packages/torch/autograd/graph.py:744: UserWarning: Plan failed with a cudnnExcep
tion: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor Failed cudnn_status: CUDNN_STATUS_NOT_
SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
    return Variable._execution_engine.run_backward( # Calls into the C++ engine to run the backward pass
79
Episode: 0: Time: 11.771679401397705 Total Reward: -62.62589928057605 Avg_Loss: 0.6153453028513284
238
Episode: 1: Time: 12.641878843307495 Total Reward: -43.71794871794934 Avg_Loss: 0.4967565799538954
238
Episode: 2: Time: 11.999724864959717 Total Reward: -76.75182481751825 Avg_Loss: 0.5925275268260107
238
Episode: 3: Time: 11.693858623504639 Total Reward: -34.2857142857148 Avg_Loss: 0.5343256173498866
238
Episode: 4: Time: 12.666028022766113 Total Reward: 2.643097643097237 Avg_Loss: 0.79901597471986
238
Episode: 5: Time: 12.367794036865234 Total Reward: -73.87323943661988 Avg_Loss: 0.7308605860198746
238
Episode: 6: Time: 11.58596682548523 Total Reward: -26.297709923664712 Avg_Loss: 0.6685909091518456
238
Episode: 7: Time: 10.986618041992188 Total Reward: -52.031250000000504 Avg_Loss: 0.796609386519975
238
Episode: 8: Time: 11.639135122299194 Total Reward: -11.666666666667503 Avg_Loss: 0.8588160825971546
238
Episode: 9: Time: 11.618962526321411 Total Reward: 71.66666666667095 Avg_Loss: 1.0187797502403249
238
Episode: 10: Time: 12.249074220657349 Total Reward: 22.6470588235303 Avg_Loss: 1.2433293081757402
238
Episode: 11: Time: 12.495743751525879 Total Reward: 258.7906137184044 Avg_Loss: 1.8668715566114968
238
Episode: 12: Time: 13.794064044952393 Total Reward: 28.86706948640626 Avg_Loss: 2.5728339117114283
238
Episode: 13: Time: 13.450676202774048 Total Reward: 184.86348122867193 Avg_Loss: 2.945577900825428
238
Episode: 14: Time: 12.922200679779053 Total Reward: 318.55932203389193 Avg_Loss: 3.5687268224834394
238
Episode: 15: Time: 13.213506937026978 Total Reward: 130.08038585209346 Avg_Loss: 3.4406628408351865
238
Episode: 16: Time: 12.541733980178833 Total Reward: 343.5964912280682 Avg_Loss: 3.6789981098610816
238
Episode: 17: Time: 12.410815477371216 Total Reward: 13.474576271187587 Avg_Loss: 3.408858998357749
238
Episode: 18: Time: 13.3681321144104 Total Reward: 119.95327102804119 Avg_Loss: 3.057307433493498
214
Episode: 19: Time: 11.642040014266968 Total Reward: 164.1145631067859 Avg_Loss: 4.301601111471096
238
Episode: 20: Time: 12.838069915771484 Total Reward: 408.52112676055503 Avg_Loss: 9.671166020656834
238
Episode: 21: Time: 13.016173839569092 Total Reward: 367.83783783783593 Avg_Loss: 7.485193159149475
238
Episode: 22: Time: 12.747554063796997 Total Reward: 323.43971631204715 Avg_Loss: 9.17123949076949
238
Episode: 23: Time: 12.796058893203735 Total Reward: 456.97132616486243 Avg_Loss: 8.177026592883744
238
Episode: 24: Time: 12.787243127822876 Total Reward: 196.6666666666708 Avg_Loss: 6.1561172977715986
238
Episode: 25: Time: 12.436172723770142 Total Reward: 205.36630036630118 Avg_Loss: 4.729692517959771
238
Episode: 26: Time: 12.298464298248291 Total Reward: 20.38461538461626 Avg_Loss: 6.044970732902279
238
Episode: 27: Time: 12.512173175811768 Total Reward: 363.0419580419556 Avg_Loss: 5.948073168011272
238
Episode: 28: Time: 12.729273319244385 Total Reward: 249.82758620688742 Avg_Loss: 4.81384256006289
238
Episode: 29: Time: 12.504530429840088 Total Reward: 644.4366197182983 Avg_Loss: 4.916794099477159

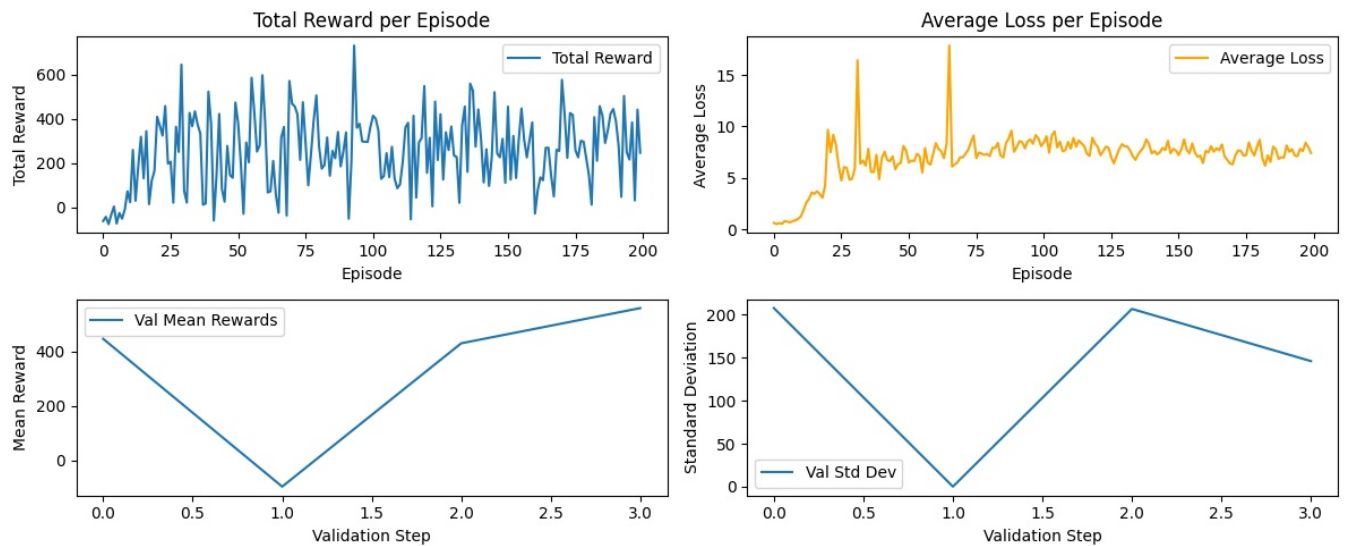
```

238
Episode: 30: Time: 12.389463424682617 Total Reward: 73.458781362007 Avg_Loss: 5.927853087667658
238
Episode: 31: Time: 12.267624139785767 Total Reward: 20.625000000001464 Avg_Loss: 16.420780912166883
238
Episode: 32: Time: 12.451112747192383 Total Reward: 425.912547528512 Avg_Loss: 6.375626707527818
238
Episode: 33: Time: 12.749916076660156 Total Reward: 365.7843137254864 Avg_Loss: 6.674003550485403
238
Episode: 34: Time: 12.618664741516113 Total Reward: 433.48101265822226 Avg_Loss: 6.178022580487387
238
Episode: 35: Time: 13.222193717956543 Total Reward: 373.55345911948774 Avg_Loss: 7.817924404344639
238
Episode: 36: Time: 12.670234680175781 Total Reward: 333.5714285714253 Avg_Loss: 5.573309904637457
238
Episode: 37: Time: 11.4294114112854 Total Reward: 10.919003115265701 Avg_Loss: 5.556101844340813
238
Episode: 38: Time: 11.486623764038086 Total Reward: 16.94029850746145 Avg_Loss: 7.234086628721542
238
Episode: 39: Time: 12.559608221054077 Total Reward: 522.0886075949311 Avg_Loss: 4.868024786480334
238
Episode: 40: Time: 12.356297731399536 Total Reward: 377.72727272726604 Avg_Loss: 6.914346668650122
238
Episode: 41: Time: 11.590977907180786 Total Reward: -60.648854961832804 Avg_Loss: 7.539970796649196
238
Episode: 42: Time: 12.6817045211792 Total Reward: 141.84210526316173 Avg_Loss: 6.739629669099295
238
Episode: 43: Time: 12.597564458847046 Total Reward: 421.3636363636307 Avg_Loss: 6.604320933588412
238
Episode: 44: Time: 12.713473558425903 Total Reward: 81.27118644068186 Avg_Loss: 7.077177289153347
238
Episode: 45: Time: 12.896863222122192 Total Reward: 24.86301369862897 Avg_Loss: 5.833955084826766
238
Episode: 46: Time: 12.593072652816772 Total Reward: 276.4285714285634 Avg_Loss: 6.328090578317642
238
Episode: 47: Time: 12.53774619102478 Total Reward: 144.61661341853372 Avg_Loss: 6.436611131960604
238
Episode: 48: Time: 12.561641693115234 Total Reward: 132.89115646258725 Avg_Loss: 8.09745152908213
238
Episode: 49: Time: 12.69307565689087 Total Reward: 473.02721088434896 Avg_Loss: 7.632421144417354
self.validation_rewards = [446.1980748641243]
Validation Mean Reward: 446.1980748641243 Validation Std Reward: 207.8616985768608
238
Episode: 50: Time: 13.376353979110718 Total Reward: 381.35135135134556 Avg_Loss: 6.484269324470969
238
Episode: 51: Time: 13.431301355361938 Total Reward: 201.6360856269127 Avg_Loss: 6.6673992176015835
238
Episode: 52: Time: 13.6064772605896 Total Reward: -30.374149659864475 Avg_Loss: 6.630002453046687
238
Episode: 53: Time: 13.453267097473145 Total Reward: 292.09677419354495 Avg_Loss: 7.333771170187397
238
Episode: 54: Time: 13.17055892944336 Total Reward: 202.79411764706276 Avg_Loss: 7.103946955764995
238
Episode: 55: Time: 13.069382667541504 Total Reward: 584.8561151079042 Avg_Loss: 5.523268322734272
238
Episode: 56: Time: 13.123049020767212 Total Reward: 431.31578947368126 Avg_Loss: 7.887426116386382
238
Episode: 57: Time: 13.149207592010498 Total Reward: 251.289752650169 Avg_Loss: 6.471390559643257
238
Episode: 58: Time: 13.326962232589722 Total Reward: 280.4512635379074 Avg_Loss: 6.285388238790657
238
Episode: 59: Time: 12.93295431137085 Total Reward: 596.4062499999925 Avg_Loss: 7.306301715995083
238
Episode: 60: Time: 13.028390407562256 Total Reward: 388.51648351647464 Avg_Loss: 8.359978211026231
238
Episode: 61: Time: 13.383518695831299 Total Reward: 66.29032258064898 Avg_Loss: 7.820345960745291
238
Episode: 62: Time: 13.421181678771973 Total Reward: 71.66666666667086 Avg_Loss: 7.582524896168909
238
Episode: 63: Time: 12.921090841293335 Total Reward: 208.88692579505482 Avg_Loss: 6.9068939074748705
238
Episode: 64: Time: 12.84510350227356 Total Reward: 56.006711409400076 Avg_Loss: 8.311805512724805
238
Episode: 65: Time: 13.282684087753296 Total Reward: -25.379746835443413 Avg_Loss: 17.84956596278343
238
Episode: 66: Time: 13.703181028366089 Total Reward: 315.2564102564085 Avg_Loss: 6.090165418737075
238
Episode: 67: Time: 13.274806261062622 Total Reward: 362.6271186440562 Avg_Loss: 6.292175758786562
238
Episode: 68: Time: 12.309535026550293 Total Reward: -38.06049822064129 Avg_Loss: 6.506295079944515
238
Episode: 69: Time: 13.021986722946167 Total Reward: 570.3061224489677 Avg_Loss: 6.987786619102254
238
Episode: 70: Time: 13.210989952087402 Total Reward: 467.7376425855441 Avg_Loss: 6.983485744780853
238
Episode: 71: Time: 13.106563806533813 Total Reward: 455.7246376811563 Avg_Loss: 7.333733553145112
238
Episode: 72: Time: 13.525025367736816 Total Reward: 419.1065830720962 Avg_Loss: 7.713965870753056
238

Episode: 73: Time: 13.384788751602173 Total Reward: 213.68167202572494 Avg_Loss: 8.454350025213065
238
Episode: 74: Time: 13.493896007537842 Total Reward: 474.444444444375 Avg_Loss: 9.108796668653728
238
Episode: 75: Time: 13.921338558197021 Total Reward: 292.54325259515485 Avg_Loss: 6.912924088099423
238
Episode: 76: Time: 13.603454113006592 Total Reward: 98.75000000000406 Avg_Loss: 7.453687257125598
238
Episode: 77: Time: 13.550244092941284 Total Reward: 239.42622950820112 Avg_Loss: 7.350994453710668
238
Episode: 78: Time: 13.24539566040039 Total Reward: 391.2068965517187 Avg_Loss: 7.2248454654918
238
Episode: 79: Time: 13.16910982131958 Total Reward: 504.9999999999918 Avg_Loss: 7.311867010192711
238
Episode: 80: Time: 13.486780166625977 Total Reward: 274.2307692307612 Avg_Loss: 7.108613792587729
238
Episode: 81: Time: 13.284813642501831 Total Reward: 174.23076923077014 Avg_Loss: 7.921279822577949
238
Episode: 82: Time: 12.798513650894165 Total Reward: 190.21126760563845 Avg_Loss: 7.890738635003066
238
Episode: 83: Time: 13.407203435897827 Total Reward: 314.9999999999981 Avg_Loss: 8.387810772206603
238
Episode: 84: Time: 13.506731986999512 Total Reward: 142.3887240356126 Avg_Loss: 7.169229090714655
238
Episode: 85: Time: 13.56374979019165 Total Reward: 254.2647058823418 Avg_Loss: 6.978913599202613
238
Episode: 86: Time: 13.83710765838623 Total Reward: 220.96091205212196 Avg_Loss: 8.373648651507722
238
Episode: 87: Time: 13.57845950126648 Total Reward: 340.0877192982433 Avg_Loss: 8.802956030148419
238
Episode: 88: Time: 13.216320753097534 Total Reward: 184.22077922078233 Avg_Loss: 9.582130096038851
238
Episode: 89: Time: 13.443761348724365 Total Reward: 250.86466165413512 Avg_Loss: 7.531171213678953
238
Episode: 90: Time: 13.84357738494873 Total Reward: 337.601880877741 Avg_Loss: 8.002904882451066
238
Episode: 91: Time: 13.350906133651733 Total Reward: -52.36434108527199 Avg_Loss: 8.566181193880674
238
Episode: 92: Time: 13.24547815322876 Total Reward: 212.69230769231086 Avg_Loss: 8.468597881433343
238
Episode: 93: Time: 13.382643461227417 Total Reward: 730.3424657534129 Avg_Loss: 7.863973988204443
238
Episode: 94: Time: 13.292299270629883 Total Reward: 358.94736842104703 Avg_Loss: 8.54221441916057
238
Episode: 95: Time: 13.003383874893188 Total Reward: 376.631205673757 Avg_Loss: 8.722950431228686
238
Episode: 96: Time: 13.151896476745605 Total Reward: 296.9753086419749 Avg_Loss: 8.307953784445754
238
Episode: 97: Time: 12.647845983505249 Total Reward: 295.8450704225368 Avg_Loss: 9.147365279558326
238
Episode: 98: Time: 13.454094171524048 Total Reward: 294.72809667673323 Avg_Loss: 8.838226257752972
238
Episode: 99: Time: 13.025087118148804 Total Reward: 361.1403508771866 Avg_Loss: 8.074552293585128
self.validation_rewards = [446.1980748641243, -94.99999999999999]
Validation Mean Reward: -94.99999999999999 Validation Std Reward: 3.750876069797784e-14
238
Episode: 100: Time: 12.456021547317505 Total Reward: 413.4745762711835 Avg_Loss: 8.418417084617776
238
Episode: 101: Time: 12.237483978271484 Total Reward: 399.7735191637606 Avg_Loss: 9.033785163354473
238
Episode: 102: Time: 12.41713833808899 Total Reward: 345.2730375426587 Avg_Loss: 7.452846363812935
238
Episode: 103: Time: 12.263581275939941 Total Reward: 127.97297297297666 Avg_Loss: 9.115118701918787
238
Episode: 104: Time: 12.599793195724487 Total Reward: 140.0993377483479 Avg_Loss: 9.509960526678743
238
Episode: 105: Time: 12.662678956985474 Total Reward: 244.2226148409933 Avg_Loss: 7.948619925174393
238
Episode: 106: Time: 13.131421566009521 Total Reward: 135.9859154929596 Avg_Loss: 8.505896461110154
238
Episode: 107: Time: 12.31568717956543 Total Reward: 272.6470588235314 Avg_Loss: 7.5691227376961905
238
Episode: 108: Time: 12.496800184249878 Total Reward: 127.97297297297368 Avg_Loss: 7.687806909825621
238
Episode: 109: Time: 12.666248798370361 Total Reward: 85.55555555555516 Avg_Loss: 8.479198859519318
238
Episode: 110: Time: 13.189982414245605 Total Reward: 103.25072886297775 Avg_Loss: 7.837802167700119
238
Episode: 111: Time: 12.392067432403564 Total Reward: 203.7012987013014 Avg_Loss: 8.869298711544326
238
Episode: 112: Time: 12.578240156173706 Total Reward: 360.17241379309837 Avg_Loss: 8.024089310349536
238
Episode: 113: Time: 12.604280471801758 Total Reward: 381.36363636363274 Avg_Loss: 8.630912244820795
238
Episode: 114: Time: 12.891371488571167 Total Reward: -55.00000000000006 Avg_Loss: 8.335201270940924
238
Episode: 115: Time: 12.865082502365112 Total Reward: 413.14332247556604 Avg_Loss: 8.077418188087078
238
Episode: 116: Time: 12.460869789123535 Total Reward: 43.18181818181834 Avg_Loss: 7.339749013175483

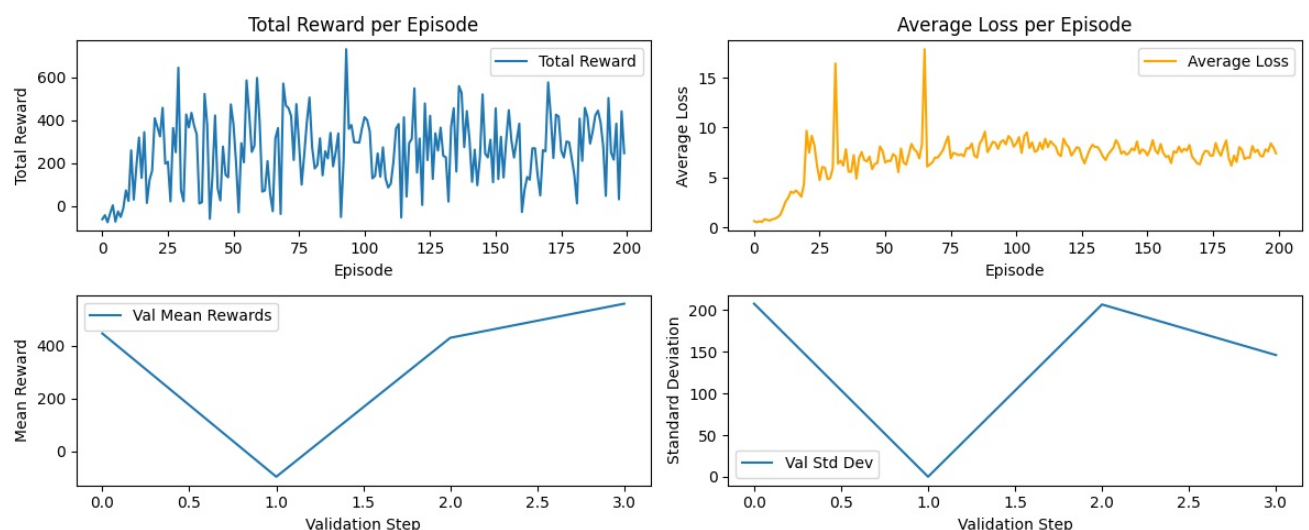
238
Episode: 117: Time: 12.822988033294678 Total Reward: 292.6221498371335 Avg_Loss: 7.130336106324396
238
Episode: 118: Time: 12.646114587783813 Total Reward: 312.5342465753414 Avg_Loss: 8.884874776631845
238
Episode: 119: Time: 12.409496307373047 Total Reward: 547.8571428571308 Avg_Loss: 8.32042945182624
238
Episode: 120: Time: 12.645491361618042 Total Reward: 155.00000000000347 Avg_Loss: 8.012476805879288
238
Episode: 121: Time: 12.890389204025269 Total Reward: 313.8235294117614 Avg_Loss: 7.190361008423717
238
Episode: 122: Time: 12.793696403503418 Total Reward: 3.726114649680613 Avg_Loss: 7.570793987823134
238
Episode: 123: Time: 12.840820550918579 Total Reward: 477.3905723905635 Avg_Loss: 8.024863133911325
238
Episode: 124: Time: 12.579308271408081 Total Reward: 213.17610062893473 Avg_Loss: 7.921133248745894
238
Episode: 125: Time: 12.611145973205566 Total Reward: 420.15151515150757 Avg_Loss: 6.983413100743494
238
Episode: 126: Time: 12.585574626922607 Total Reward: 125.18348623853274 Avg_Loss: 6.4006483775227005
238
Episode: 127: Time: 12.787795543670654 Total Reward: 338.12101910827624 Avg_Loss: 7.135557954050913
238
Episode: 128: Time: 12.543317317962646 Total Reward: 258.1353135313543 Avg_Loss: 7.8205995038777845
238
Episode: 129: Time: 12.464645624160767 Total Reward: 364.64912280700764 Avg_Loss: 8.265887251421184
238
Episode: 130: Time: 12.367780447006226 Total Reward: 233.6219081272104 Avg_Loss: 8.06014950886494
238
Episode: 131: Time: 12.646653175354004 Total Reward: 226.31147540984037 Avg_Loss: 8.034386609281812
238
Episode: 132: Time: 12.447102069854736 Total Reward: 19.64968152866161 Avg_Loss: 7.634976941998265
238
Episode: 133: Time: 12.321520328521729 Total Reward: 367.89752650176507 Avg_Loss: 7.089331545248753
238
Episode: 134: Time: 12.517628192901611 Total Reward: 455.17301038061885 Avg_Loss: 6.7459644777935095
238
Episode: 135: Time: 11.90821123123169 Total Reward: 159.6816479400767 Avg_Loss: 7.294897526753049
238
Episode: 136: Time: 11.926968812942505 Total Reward: 558.5433070866026 Avg_Loss: 7.629562124484727
238
Episode: 137: Time: 11.502682447433472 Total Reward: 526.5139442230961 Avg_Loss: 7.970379572956502
238
Episode: 138: Time: 12.415171384811401 Total Reward: 273.9024390243917 Avg_Loss: 8.731021338150281
238
Episode: 139: Time: 12.489465713500977 Total Reward: 441.8098159509166 Avg_Loss: 8.27246489144173
238
Episode: 140: Time: 13.162744045257568 Total Reward: 310.555555555551 Avg_Loss: 7.377637078782089
238
Episode: 141: Time: 12.043281555175781 Total Reward: 111.89655172413887 Avg_Loss: 7.605031248401193
238
Episode: 142: Time: 12.263861417770386 Total Reward: 261.6666666666704 Avg_Loss: 7.277529424979907
238
Episode: 143: Time: 13.290970087051392 Total Reward: 96.04477611940443 Avg_Loss: 7.463413910705502
238
Episode: 144: Time: 12.025928497314453 Total Reward: 227.03389830508706 Avg_Loss: 7.90038289993751
238
Episode: 145: Time: 11.907013416290283 Total Reward: 519.7540983606461 Avg_Loss: 7.700163132002373
238
Episode: 146: Time: 12.969353199005127 Total Reward: 243.23529411765043 Avg_Loss: 8.58934630966988
238
Episode: 147: Time: 12.597661972045898 Total Reward: 225.51282051282396 Avg_Loss: 7.379188557632831
238
Episode: 148: Time: 12.614179134368896 Total Reward: 308.3333333333485 Avg_Loss: 7.808087383498664
238
Episode: 149: Time: 12.354336738586426 Total Reward: 109.94699646643522 Avg_Loss: 7.644119143486023
self.validation_rewards = [446.1980748641243, -94.999999999999, 429.784006803002]
Validation Mean Reward: 429.784006803002 Validation Std Reward: 206.90174680528145
238
Episode: 150: Time: 12.713046073913574 Total Reward: 454.4880546075003 Avg_Loss: 7.206988518979369
238
Episode: 151: Time: 12.423624038696289 Total Reward: 124.69696969697094 Avg_Loss: 7.781530024123793
238
Episode: 152: Time: 12.885177373886108 Total Reward: 321.90962099125335 Avg_Loss: 8.726493434745725
238
Episode: 153: Time: 12.406743049621582 Total Reward: 132.1062271062306 Avg_Loss: 7.694394831396952
238
Episode: 154: Time: 12.608903884887695 Total Reward: 304.9999999999977 Avg_Loss: 7.3849784005589845
238
Episode: 155: Time: 12.644919872283936 Total Reward: 446.09589041095126 Avg_Loss: 8.338287260852942
238
Episode: 156: Time: 12.37580943107605 Total Reward: 304.25373134327924 Avg_Loss: 7.465009919234684
238
Episode: 157: Time: 12.908956289291382 Total Reward: 225.00000000000364 Avg_Loss: 7.040848477047031
238
Episode: 158: Time: 12.283987760543823 Total Reward: 302.16312056737405 Avg_Loss: 7.1550272244866155
238
Episode: 159: Time: 12.206351041793823 Total Reward: 382.94117647058505 Avg_Loss: 6.426907368066932
238

Episode: 160: Time: 11.958341360092163 Total Reward: -29.256055363322467 Avg_Loss: 7.61329898313314
238
Episode: 161: Time: 12.705008029937744 Total Reward: 73.91891891891818 Avg_Loss: 7.4655250801759605
238
Episode: 162: Time: 12.611740350723267 Total Reward: 134.7297297297336 Avg_Loss: 8.073908732217902
238
Episode: 163: Time: 12.552317142486572 Total Reward: 121.66666666667112 Avg_Loss: 7.534458415848868
238
Episode: 164: Time: 12.515504837036133 Total Reward: 268.32179930795917 Avg_Loss: 7.870245271370191
238
Episode: 165: Time: 13.026148796081543 Total Reward: 268.07692307692525 Avg_Loss: 7.683074176812372
238
Episode: 166: Time: 12.816311359405518 Total Reward: 141.76012461059395 Avg_Loss: 8.24146510122203
238
Episode: 167: Time: 12.492801189422607 Total Reward: 48.34470989761416 Avg_Loss: 7.074405008003492
238
Episode: 168: Time: 12.368876695632935 Total Reward: 259.51505016722814 Avg_Loss: 6.750620624097455
238
Episode: 169: Time: 11.9761381149292 Total Reward: 254.31506849315403 Avg_Loss: 6.402876169240775
238
Episode: 170: Time: 11.537126779556274 Total Reward: 575.731707317061 Avg_Loss: 6.306319367985766
238
Episode: 171: Time: 12.816943168640137 Total Reward: 418.35311572699953 Avg_Loss: 7.200187542859246
238
Episode: 172: Time: 11.861066341400146 Total Reward: 222.85714285714465 Avg_Loss: 7.65736982802383
238
Episode: 173: Time: 12.111982822418213 Total Reward: 425.29520295202144 Avg_Loss: 7.624049813306632
238
Episode: 174: Time: 12.072385311126709 Total Reward: 416.9047619047583 Avg_Loss: 7.168925029890878
238
Episode: 175: Time: 12.657940864562988 Total Reward: 257.0599250936251 Avg_Loss: 7.200104549652388
238
Episode: 176: Time: 12.539243698120117 Total Reward: 225.28469750890113 Avg_Loss: 8.439777660770577
238
Episode: 177: Time: 12.443247556686401 Total Reward: 299.83394833947443 Avg_Loss: 7.631029678993866
238
Episode: 178: Time: 12.562392950057983 Total Reward: 296.3043478260884 Avg_Loss: 7.17019328850658
238
Episode: 179: Time: 12.83012318611145 Total Reward: 221.83168316832115 Avg_Loss: 7.9872752277791
238
Episode: 180: Time: 12.736872434616089 Total Reward: 143.24451410658398 Avg_Loss: 8.704905499430264
238
Episode: 181: Time: 12.354028940200806 Total Reward: 11.16438356164451 Avg_Loss: 6.930808414932058
238
Episode: 182: Time: 12.465658187866211 Total Reward: 406.7182130584137 Avg_Loss: 6.174667070893681
238
Episode: 183: Time: 13.096981287002563 Total Reward: 209.59770114942916 Avg_Loss: 7.176832367392147
238
Episode: 184: Time: 12.39993953704834 Total Reward: 456.4403292180989 Avg_Loss: 6.553116969701622
238
Episode: 185: Time: 12.523475170135498 Total Reward: 414.22509225091665 Avg_Loss: 8.035841306718458
238
Episode: 186: Time: 13.167201280593872 Total Reward: 290.35031847133547 Avg_Loss: 7.770027023403585
238
Episode: 187: Time: 13.199151754379272 Total Reward: 347.424242424239 Avg_Loss: 6.835813367567143
238
Episode: 188: Time: 12.610134840011597 Total Reward: 422.3611111111033 Avg_Loss: 7.002224967259319
238
Episode: 189: Time: 12.68550419807434 Total Reward: 443.7323943661904 Avg_Loss: 6.95974413336826
238
Episode: 190: Time: 12.340453386306763 Total Reward: 388.7545126353707 Avg_Loss: 8.143194683960505
238
Episode: 191: Time: 12.69851279258728 Total Reward: 278.28767123287656 Avg_Loss: 7.523727703495186
238
Episode: 192: Time: 12.288314819335938 Total Reward: 46.50943396226737 Avg_Loss: 7.7698365179430535
238
Episode: 193: Time: 12.334615230560303 Total Reward: 502.74436090224384 Avg_Loss: 7.16707217593153
238
Episode: 194: Time: 12.760040044784546 Total Reward: 248.65325077399592 Avg_Loss: 7.091197771685464
238
Episode: 195: Time: 12.391777515411377 Total Reward: 215.34482758621064 Avg_Loss: 7.813392955215037
238
Episode: 196: Time: 12.530243158340454 Total Reward: 383.1021897810176 Avg_Loss: 7.603496357172477
238
Episode: 197: Time: 12.90114951133728 Total Reward: 30.35612535612485 Avg_Loss: 8.406632016686832
238
Episode: 198: Time: 12.401681900024414 Total Reward: 440.5805243445665 Avg_Loss: 7.997318063463483
238
Episode: 199: Time: 12.80248737335205 Total Reward: 245.76433121019076 Avg_Loss: 7.405968129133978
self.validation_rewards = [446.1980748641243, -94.999999999999, 429.784006803002, 558.6625393003231]
Validation Mean Reward: 558.6625393003231 Validation Std Reward: 146.13785385264396
Test Mean Reward: 596.4705857600395 Test Std Reward: 97.0498457000493



Please include a plot of the training and validation rewards over the episodes in the report. An additional question to answer is does the loss matter in DQN? Why or why not?

We can also draw a animation of the car in one game, the code is provided below



Does the loss plot matter?

Our understanding is that loss plot might not matter as much for value-based learning approaches like DQN in RL, for the following reasons:

- **Reward is the Primary Metric:**

In reinforcement learning, the ultimate goal is to maximize the cumulative reward, not necessarily to minimize the loss. Sometimes a model might achieve high rewards even if the loss plot is not perfectly smooth or continuously decreasing.

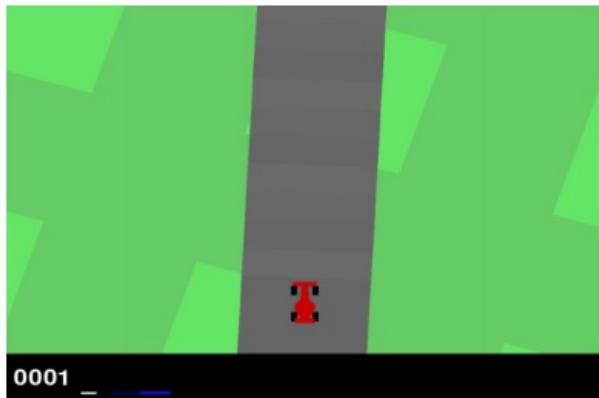
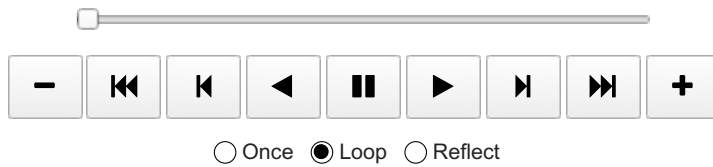
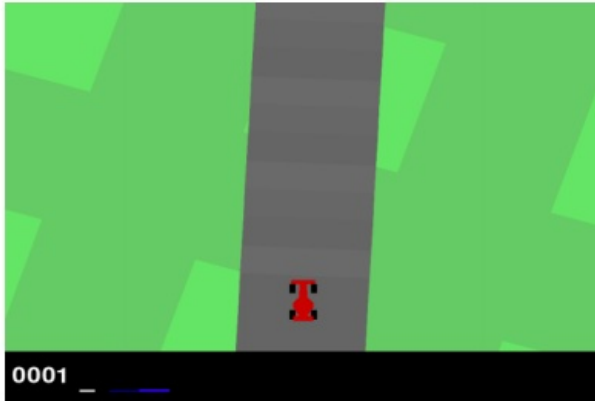
- **Exploration vs. Exploitation:**

During the exploration phase, the loss might not decrease steadily due to the agent trying out new actions. This exploration is essential for finding the optimal policy, even if it temporarily increases the loss.

```
In [ ]: eval_env = gym.make('CarRacing-v2', continuous=True, render_mode='rgb_array')
eval_env = EnvWrapper(eval_env)

total_rewards, frames = trainerDQN.play_episode(0, True, 42)
anim = animate(frames)
HTML(anim.to_jshtml())
```

Out[]:



Double DQN

In the original paper, where the algorithm is shown above, the estimated target Q value was computed using the current Q network's weights. However, this can lead to overestimation of the Q values. To mitigate this, we can use the target network to compute the target Q value. This is known as Double DQN.

Hard updating Target Network (5 points)

Original implementations for this involved hard updates, where the model weights were copied to the target network every C steps. This is known as hard updating. This was what was used in the Nature Paper by Mnih et al 2015 "Human-level control through deep reinforcement learning"

Please implement this by implementing the `optimize model` and `update model` classes in `HardUpdateDQN` in `DQN.py`.

[illegible]

```
device = 'cpu',
seed = 42,
epsilon_scheduler = utils.exponential_decay(1, 1000, 0.1),
save_path = utils.get_save_path("DoubleDQN_HardUpdates/", "./runs/")
```

```
trainerHardUpdateDQN.train(200, 50, 30, 50, 50)
```

saving to ./runs/DoubleDQN_HardUpdates/run13

```
/content/drive/MyDrive/DQN_Project4/DQN.py:370: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
```

```
states = torch.tensor(states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:371: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
```

```
actions = torch.tensor(actions).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:372: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
```

```
rewards = torch.tensor(rewards).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:373: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
```

```
next_states = torch.tensor(next_states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:374: UserWarning: To copy construct from a tensor, it is recommended
to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.t
ensor(sourceTensor).
```

```
dones = torch.tensor(dones).clone().detach().float().to(self.device)
```

79

Episode: 0: Time: 20.49198007583618 Total Reward: -55.43165467625943 Avg_Loss: 0.7018106116712848

238

Episode: 1: Time: 33.71599364280701 Total Reward: -43.717948717948914 Avg_Loss: 0.622744302719268

238

Episode: 2: Time: 32.09483313560486 Total Reward: -22.007299270072878 Avg_Loss: 0.7135957754412744

238

Episode: 3: Time: 33.920278787612915 Total Reward: -70.0000000000004 Avg_Loss: 0.7073146083994823

238

Episode: 4: Time: 33.981526374816895 Total Reward: -44.49494949494981 Avg_Loss: 0.6876790711045766

238

Episode: 5: Time: 34.652745485305786 Total Reward: -21.056338028168845 Avg_Loss: 0.733820206008288

238

Episode: 6: Time: 34.68365001678467 Total Reward: -30.114503816794485 Avg_Loss: 0.7282387454907934

238

Episode: 7: Time: 33.41555142402649 Total Reward: -79.37499999999989 Avg_Loss: 0.6661832616781863

238

Episode: 8: Time: 37.534449338912964 Total Reward: -37.02898550724707 Avg_Loss: 0.812886790849832

238

Episode: 9: Time: 33.58968687057495 Total Reward: -38.18181818181887 Avg_Loss: 0.7199338083742421

238

Episode: 10: Time: 34.091148138046265 Total Reward: -52.51633986928145 Avg_Loss: 0.7376436289212033

204

Episode: 11: Time: 31.01764750480652 Total Reward: -33.185559566784065 Avg_Loss: 0.8754805767835647

238

Episode: 12: Time: 36.18769574165344 Total Reward: -28.53474320241759 Avg_Loss: 0.9028155937196076

238

Episode: 13: Time: 36.25651955604553 Total Reward: 14.215017064845586 Avg_Loss: 0.9637796022624028

238

Episode: 14: Time: 35.3432834148407 Total Reward: -33.983050847458316 Avg_Loss: 0.9482087235929084

238

Episode: 15: Time: 37.607983350753784 Total Reward: -30.691318327974972 Avg_Loss: 1.0079525755170513

238

Episode: 16: Time: 35.1061327457428 Total Reward: 27.80701754386125 Avg_Loss: 3.6979145301161567

238

Episode: 17: Time: 36.66626596450806 Total Reward: 13.474576271187146 Avg_Loss: 3.378598066250316

238

Episode: 18: Time: 34.64903926849365 Total Reward: -38.925233644860256 Avg_Loss: 1.4111841660337288

173

Episode: 19: Time: 26.69685882904053 Total Reward: -55.83139158576039 Avg_Loss: 1.1779206645368152

238

Episode: 20: Time: 35.8691086769104 Total Reward: 292.3239436619688 Avg_Loss: 8.997693459276391

238

Episode: 21: Time: 35.95502758026123 Total Reward: 337.4324324324246 Avg_Loss: 1.6681221867559337

238

Episode: 22: Time: 37.07455801963806 Total Reward: -27.624113475177893 Avg_Loss: 10.552304851345154

238

Episode: 23: Time: 34.752267360687256 Total Reward: 26.863799283156254 Avg_Loss: 1.7568906166849017

238

Episode: 24: Time: 39.64921283721924 Total Reward: 206.28205128205536 Avg_Loss: 4.556486487920795

238

Episode: 25: Time: 35.53161811828613 Total Reward: 220.018315018318 Avg_Loss: 8.372161120176315

238

Episode: 26: Time: 37.17413067817688 Total Reward: 499.40559440558786 Avg_Loss: 2.2169344741694568

238

Episode: 27: Time: 34.358871936798096 Total Reward: -32.06293706293778 Avg_Loss: 2.8465557288722834

238

Episode: 28: Time: 38.18307328224182 Total Reward: 234.15360501567562 Avg_Loss: 2.470015037836147

238

Episode: 29: Time: 36.562358379364014 Total Reward: 169.0845070422583 Avg_Loss: 2.9192953287553385

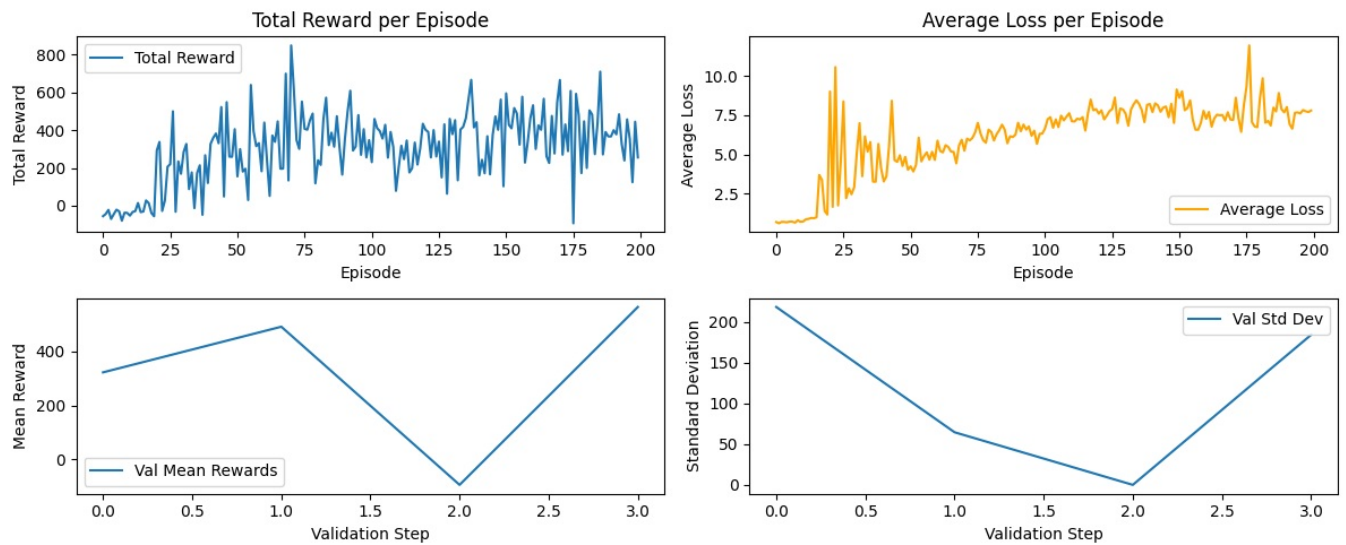
238

Episode: 30: Time: 36.06045436859131 Total Reward: 284.9283154121783 Avg_Loss: 5.177030552836025
238
Episode: 31: Time: 35.570146799087524 Total Reward: 326.8749999999934 Avg_Loss: 6.989830666730384
238
Episode: 32: Time: 37.100969314575195 Total Reward: 87.50950570342664 Avg_Loss: 3.616307547237693
238
Episode: 33: Time: 35.76886606216431 Total Reward: 176.24183006536416 Avg_Loss: 6.145894882871824
238
Episode: 34: Time: 37.62418842315674 Total Reward: -12.721518987342309 Avg_Loss: 5.181626982799097
238
Episode: 35: Time: 37.047890424728394 Total Reward: 172.29559748428116 Avg_Loss: 5.786431454184677
238
Episode: 36: Time: 37.80131912231445 Total Reward: 213.97009966777867 Avg_Loss: 3.273620235944996
238
Episode: 37: Time: 36.77357292175293 Total Reward: -48.271028037383886 Avg_Loss: 3.264705375400411
238
Episode: 38: Time: 35.449936389923096 Total Reward: 266.94029850745824 Avg_Loss: 5.668330505740743
238
Episode: 39: Time: 37.54961609840393 Total Reward: 120.1898734177255 Avg_Loss: 4.124399456406842
238
Episode: 40: Time: 35.81584310531616 Total Reward: 326.81818181817573 Avg_Loss: 3.284165035025412
238
Episode: 41: Time: 36.72558784484863 Total Reward: 359.19847328244373 Avg_Loss: 3.609064592533753
238
Episode: 42: Time: 37.63911294937134 Total Reward: 381.9736842105141 Avg_Loss: 5.566217866514911
238
Episode: 43: Time: 37.116084814071655 Total Reward: 330.4545454545471 Avg_Loss: 8.409792219891267
238
Episode: 44: Time: 36.0777325630188 Total Reward: 521.9491525423626 Avg_Loss: 4.638890436467002
238
Episode: 45: Time: 37.35290837287903 Total Reward: 48.835616438358215 Avg_Loss: 4.527769481434541
238
Episode: 46: Time: 35.76896929740906 Total Reward: 547.8571428571353 Avg_Loss: 4.94820198316534
238
Episode: 47: Time: 37.12446117401123 Total Reward: 259.6325878594275 Avg_Loss: 4.2674440618823555
238
Episode: 48: Time: 35.65023112297058 Total Reward: 258.7414965986387 Avg_Loss: 4.845822474786213
238
Episode: 49: Time: 37.77707505226135 Total Reward: 404.9999999999967 Avg_Loss: 4.030237199128175
self.validation_rewards = [322.66424841618806]
Validation Mean Reward: 322.66424841618806 Validation Std Reward: 218.58558293758372
238
Episode: 50: Time: 37.46600413322449 Total Reward: 155.00000000000443 Avg_Loss: 4.240714834768231
238
Episode: 51: Time: 39.56397771835327 Total Reward: 299.49541284403614 Avg_Loss: 3.9161600282713143
238
Episode: 52: Time: 35.80368375778198 Total Reward: 180.5102040816357 Avg_Loss: 4.3392797089925335
238
Episode: 53: Time: 37.13311839103699 Total Reward: 195.3225806451656 Avg_Loss: 6.068634971475401
238
Episode: 54: Time: 35.49104571342468 Total Reward: 29.999999999999098 Avg_Loss: 4.560496360063553
238
Episode: 55: Time: 37.156123876571655 Total Reward: 638.8129496402769 Avg_Loss: 4.892040882791791
238
Episode: 56: Time: 35.77096486091614 Total Reward: 394.87854251011487 Avg_Loss: 5.132589653760445
238
Episode: 57: Time: 37.58759641647339 Total Reward: 314.8939929328572 Avg_Loss: 4.665657635246005
238
Episode: 58: Time: 38.20551824569702 Total Reward: 330.9927797833847 Avg_Loss: 5.17029262640897
238
Episode: 59: Time: 36.93869972229004 Total Reward: 182.34375000000466 Avg_Loss: 4.651262453123301
238
Episode: 60: Time: 36.690746784210205 Total Reward: 439.79853479852915 Avg_Loss: 5.877292759027801
238
Episode: 61: Time: 36.14127540588379 Total Reward: 240.48387096774584 Avg_Loss: 5.216547372962246
238
Episode: 62: Time: 37.48652911186218 Total Reward: 51.66666666666978 Avg_Loss: 5.133854829964518
238
Episode: 63: Time: 36.372180700302124 Total Reward: 371.43109540635857 Avg_Loss: 5.591568306464107
238
Episode: 64: Time: 37.24811768531799 Total Reward: 337.88590604026405 Avg_Loss: 5.4742519910595995
238
Episode: 65: Time: 38.805683851242065 Total Reward: 446.139240506322 Avg_Loss: 5.187592871048871
238
Episode: 66: Time: 37.214128732681274 Total Reward: 195.59829059829488 Avg_Loss: 5.159935070436542
238
Episode: 67: Time: 36.90664839744568 Total Reward: 196.52542372881777 Avg_Loss: 4.436214825185407
238
Episode: 68: Time: 35.52813172340393 Total Reward: 698.5943060498096 Avg_Loss: 5.562032892924397
238
Episode: 69: Time: 36.40731191635132 Total Reward: 133.57142857143282 Avg_Loss: 5.934697260876663
238
Episode: 70: Time: 35.9976646900177 Total Reward: 847.9657794676627 Avg_Loss: 5.238873672585528
238
Episode: 71: Time: 37.401564598083496 Total Reward: 615.1449275362231 Avg_Loss: 6.002631612685549
238
Episode: 72: Time: 40.16049814224243 Total Reward: 347.0062695924743 Avg_Loss: 5.895704981158762
238
Episode: 73: Time: 37.39699602127075 Total Reward: 300.4983922829598 Avg_Loss: 6.053786674467456

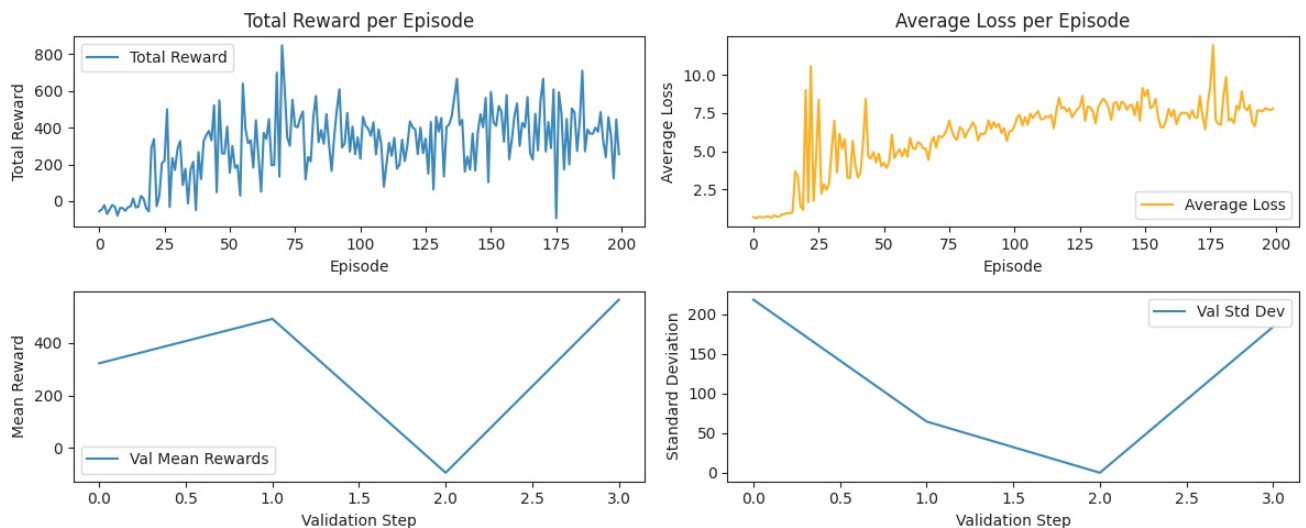
238
Episode: 74: Time: 37.32679581642151 Total Reward: 550.833333333323 Avg_Loss: 6.397007982019617
238
Episode: 75: Time: 36.79253029823303 Total Reward: 406.7301038062186 Avg_Loss: 7.007190763449469
238
Episode: 76: Time: 39.15527319908142 Total Reward: 401.8749999999932 Avg_Loss: 6.33229573859888
238
Episode: 77: Time: 37.17736101150513 Total Reward: 452.5409836065502 Avg_Loss: 5.902907852866068
238
Episode: 78: Time: 38.260186195373535 Total Reward: 487.7586206896449 Avg_Loss: 5.746504410475242
238
Episode: 79: Time: 37.72957468032837 Total Reward: 118.33333333333754 Avg_Loss: 6.575212137538846
238
Episode: 80: Time: 36.33753752708435 Total Reward: 240.3846153846131 Avg_Loss: 6.42559317680968
238
Episode: 81: Time: 37.69892954826355 Total Reward: 215.89743589743964 Avg_Loss: 5.885620526405943
238
Episode: 82: Time: 36.78970718383789 Total Reward: 457.81690140844296 Avg_Loss: 6.289096274796655
238
Episode: 83: Time: 36.57883167266846 Total Reward: 571.6666666666567 Avg_Loss: 6.5452516349423835
238
Episode: 84: Time: 38.20972156524658 Total Reward: 320.43026706231353 Avg_Loss: 6.892931020810824
238
Episode: 85: Time: 37.96257925033569 Total Reward: 386.6176470588215 Avg_Loss: 6.582856529901008
238
Episode: 86: Time: 37.932446002960205 Total Reward: 312.1661237784996 Avg_Loss: 5.7113046691197304
238
Episode: 87: Time: 36.4220495223999 Total Reward: 473.42105263157225 Avg_Loss: 6.155395335509997
238
Episode: 88: Time: 37.394131660461426 Total Reward: 307.59740259740045 Avg_Loss: 6.10724509213151
238
Episode: 89: Time: 37.928343534469604 Total Reward: 164.3984962406059 Avg_Loss: 6.251528812055828
238
Episode: 90: Time: 36.63145303726196 Total Reward: 353.2758620689635 Avg_Loss: 7.015751254658739
238
Episode: 91: Time: 37.66501212120056 Total Reward: 498.02325581394865 Avg_Loss: 6.477191504811039
238
Episode: 92: Time: 37.952754974365234 Total Reward: 608.2967032966926 Avg_Loss: 6.899174305571227
238
Episode: 93: Time: 36.7457160949707 Total Reward: 291.98630136985776 Avg_Loss: 6.581223937643676
238
Episode: 94: Time: 37.93951225280762 Total Reward: 312.89473684210384 Avg_Loss: 6.778577791542566
238
Episode: 95: Time: 36.465107917785645 Total Reward: 479.46808510637476 Avg_Loss: 6.190624835611391
238
Episode: 96: Time: 38.27849555015564 Total Reward: 269.1975308642009 Avg_Loss: 6.507628451375401
238
Episode: 97: Time: 36.492682456970215 Total Reward: 404.9999999999716 Avg_Loss: 5.671159239376292
238
Episode: 98: Time: 37.7072222328186 Total Reward: 255.4531722054358 Avg_Loss: 6.31139626472938
238
Episode: 99: Time: 40.081907749176025 Total Reward: 347.1052631578865 Avg_Loss: 6.327430366468029
self.validation_rewards = [322.66424841618806, 492.1341337621951]
Validation Mean Reward: 492.1341337621951 Validation Std Reward: 64.65416705135303
238
Episode: 100: Time: 37.56429862976074 Total Reward: 230.42372881356232 Avg_Loss: 6.647948443388739
238
Episode: 101: Time: 36.1541383266449 Total Reward: 459.0069686411088 Avg_Loss: 7.243626763840683
238
Episode: 102: Time: 37.926703453063965 Total Reward: 410.11945392490696 Avg_Loss: 7.367756741387503
238
Episode: 103: Time: 36.31591582298279 Total Reward: 394.864864864854 Avg_Loss: 6.6990493596101
238
Episode: 104: Time: 37.962578773498535 Total Reward: 355.33112582781246 Avg_Loss: 7.225199452969206
238
Episode: 105: Time: 37.12783098220825 Total Reward: 427.96819787985515 Avg_Loss: 6.739232024725745
238
Episode: 106: Time: 38.78490328788757 Total Reward: 254.29577464789048 Avg_Loss: 7.464906148549889
238
Episode: 107: Time: 37.85580229759216 Total Reward: 390.2941176470558 Avg_Loss: 7.16313516642867
238
Episode: 108: Time: 34.383360862731934 Total Reward: 310.405405405404 Avg_Loss: 7.4050830963279015
238
Episode: 109: Time: 35.97472643852234 Total Reward: 78.61111111111546 Avg_Loss: 7.619692118227983
238
Episode: 110: Time: 35.06687068939209 Total Reward: 202.37609329446263 Avg_Loss: 7.109497104372297
238
Episode: 111: Time: 36.03481650352478 Total Reward: 317.3376623376565 Avg_Loss: 7.092171530763642
238
Episode: 112: Time: 34.48043990135193 Total Reward: 246.3793103448317 Avg_Loss: 7.272509729661861
238
Episode: 113: Time: 37.755236864089966 Total Reward: 344.9999999999443 Avg_Loss: 7.220651077623127
238
Episode: 114: Time: 34.88873338699341 Total Reward: 175.7692307692346 Avg_Loss: 7.372840731584725
238
Episode: 115: Time: 36.11512017250061 Total Reward: 198.15960912052566 Avg_Loss: 6.5133629581507515
238
Episode: 116: Time: 34.520363569259644 Total Reward: 334.09090909090673 Avg_Loss: 7.630664997741956
238

Episode: 117: Time: 34.758408069610596 Total Reward: 217.70358306189178 Avg_Loss: 8.48889125244958
238
Episode: 118: Time: 35.23270058631897 Total Reward: 302.2602739726047 Avg_Loss: 7.806896593390393
238
Episode: 119: Time: 33.909334659576416 Total Reward: 433.57142857141946 Avg_Loss: 7.883215618233721
238
Episode: 120: Time: 37.470433950424194 Total Reward: 401.7105263157762 Avg_Loss: 7.596054643142123
238
Episode: 121: Time: 34.7586452960968 Total Reward: 390.29411764705213 Avg_Loss: 7.802069227234656
238
Episode: 122: Time: 36.571815967559814 Total Reward: 255.3184713375835 Avg_Loss: 7.227696906117832
238
Episode: 123: Time: 34.70050024986267 Total Reward: 399.94949494948855 Avg_Loss: 7.466701431434696
238
Episode: 124: Time: 35.764533281326294 Total Reward: 260.3459119496892 Avg_Loss: 7.75338599661819
238
Episode: 125: Time: 34.303353786468506 Total Reward: 339.34343434343305 Avg_Loss: 7.8871846469510505
238
Episode: 126: Time: 36.06828737258911 Total Reward: 149.648318042818 Avg_Loss: 8.616923803040962
238
Episode: 127: Time: 36.29914307594299 Total Reward: 430.47770700636335 Avg_Loss: 6.997672695071757
238
Episode: 128: Time: 36.10760736465454 Total Reward: 63.41584158415722 Avg_Loss: 7.917608879694418
238
Episode: 129: Time: 34.32688117027283 Total Reward: 459.3859649122707 Avg_Loss: 7.862897026939552
238
Episode: 130: Time: 35.371402740478516 Total Reward: 378.49823321554453 Avg_Loss: 7.5896157807662705
238
Episode: 131: Time: 34.82022404670715 Total Reward: 452.54098360655126 Avg_Loss: 6.822316797841497
238
Episode: 132: Time: 35.53267240524292 Total Reward: 134.2993630573286 Avg_Loss: 7.854712995661407
238
Episode: 133: Time: 34.21723484992981 Total Reward: 403.2332155476973 Avg_Loss: 8.190968200439164
238
Episode: 134: Time: 36.18705773353577 Total Reward: 417.1107266435975 Avg_Loss: 8.439287572848697
238
Episode: 135: Time: 35.73236918449402 Total Reward: 463.0524344569251 Avg_Loss: 8.211308753790975
238
Episode: 136: Time: 34.57933950424194 Total Reward: 566.4173228346378 Avg_Loss: 7.823751187625051
238
Episode: 137: Time: 35.504207134246826 Total Reward: 665.9561752987943 Avg_Loss: 7.054994515010288
238
Episode: 138: Time: 35.311856269836426 Total Reward: 414.1463414634102 Avg_Loss: 8.153500420706612
238
Episode: 139: Time: 36.702553510665894 Total Reward: 441.80981595091305 Avg_Loss: 8.205969815494633
238
Episode: 140: Time: 35.07664442062378 Total Reward: 160.55555555555884 Avg_Loss: 7.691531738313306
238
Episode: 141: Time: 37.688262939453125 Total Reward: 242.93103448276256 Avg_Loss: 8.24728565957366
238
Episode: 142: Time: 34.58274221420288 Total Reward: 171.66666666667086 Avg_Loss: 8.121066403990032
238
Episode: 143: Time: 37.067349433898926 Total Reward: 367.6865671641755 Avg_Loss: 7.712272196256814
238
Episode: 144: Time: 34.00811767578125 Total Reward: 166.01694915254222 Avg_Loss: 7.9814768378474135
238
Episode: 145: Time: 35.62365102767944 Total Reward: 376.31147540983227 Avg_Loss: 8.045703665549015
238
Episode: 146: Time: 35.49102020263672 Total Reward: 472.6470588235233 Avg_Loss: 7.365357769136669
238
Episode: 147: Time: 36.21136975288391 Total Reward: 401.79487179486756 Avg_Loss: 8.216966785803562
238
Episode: 148: Time: 36.43265891075134 Total Reward: 561.6666666666558 Avg_Loss: 7.0033216972311
238
Episode: 149: Time: 35.707297801971436 Total Reward: 102.87985865724447 Avg_Loss: 9.129275305932309
self.validation_rewards = [322.66424841618806, 492.1341337621951, -94.999999999999]
Validation Mean Reward: -94.999999999999 Validation Std Reward: 3.188218266902034e-14
238
Episode: 150: Time: 36.14198565483093 Total Reward: 594.4197952218296 Avg_Loss: 8.592745430830146
238
Episode: 151: Time: 34.530046701431274 Total Reward: 427.7272727272668 Avg_Loss: 9.013120413327417
238
Episode: 152: Time: 36.87309241294861 Total Reward: 409.3731778425611 Avg_Loss: 7.8069148223941065
238
Episode: 153: Time: 34.23028898239136 Total Reward: 516.721611721606 Avg_Loss: 7.9465717042193695
238
Episode: 154: Time: 36.122597455978394 Total Reward: 487.75862068964307 Avg_Loss: 8.434566511827356
238
Episode: 155: Time: 34.33213496208191 Total Reward: 322.80821917807106 Avg_Loss: 7.197698287603234
238
Episode: 156: Time: 37.36003518104553 Total Reward: 576.6417910447663 Avg_Loss: 6.570310094777276
238
Episode: 157: Time: 35.10496377944946 Total Reward: 228.0769230769268 Avg_Loss: 6.572109782896122
238
Episode: 158: Time: 34.243847608566284 Total Reward: 334.0780141843881 Avg_Loss: 7.0128459259241565
238
Episode: 159: Time: 35.357911109924316 Total Reward: 463.82352941175657 Avg_Loss: 7.772526530658498
238
Episode: 160: Time: 35.55266976356506 Total Reward: 531.297577854665 Avg_Loss: 7.265663636832678

238
Episode: 161: Time: 35.56881618499756 Total Reward: 300.27027027026304 Avg_Loss: 7.701206333496991
238
Episode: 162: Time: 34.75125527381897 Total Reward: 425.27027027026736 Avg_Loss: 6.762860922252431
238
Episode: 163: Time: 37.00278306007385 Total Reward: 401.66666666666174 Avg_Loss: 7.292031231046725
238
Episode: 164: Time: 35.2750358581543 Total Reward: 565.8996539792311 Avg_Loss: 7.525159063960324
238
Episode: 165: Time: 35.646687269210815 Total Reward: 261.92307692307895 Avg_Loss: 7.488990311863041
238
Episode: 166: Time: 35.554537534713745 Total Reward: 225.87227414330385 Avg_Loss: 7.5071704067102
238
Episode: 167: Time: 35.29639983177185 Total Reward: 474.96587030715574 Avg_Loss: 7.149989847375565
238
Episode: 168: Time: 36.59074378013611 Total Reward: 276.2374581939717 Avg_Loss: 7.679249199498601
238
Episode: 169: Time: 35.06678819656372 Total Reward: 548.8356164383451 Avg_Loss: 7.220740847226952
238
Episode: 170: Time: 38.37865090370178 Total Reward: 665.1626016260092 Avg_Loss: 7.187946187347925
238
Episode: 171: Time: 35.70889472961426 Total Reward: 269.9851632047423 Avg_Loss: 8.605038930888938
238
Episode: 172: Time: 36.31606936454773 Total Reward: 429.99999999999443 Avg_Loss: 7.285244757888698
238
Episode: 173: Time: 34.43712615966797 Total Reward: 288.7638376383702 Avg_Loss: 6.439335775475542
238
Episode: 174: Time: 36.12956428527832 Total Reward: 607.380952380942 Avg_Loss: 8.133127903737941
171
Episode: 175: Time: 23.585628032684326 Total Reward: -93.19363295880227 Avg_Loss: 9.334395237833435
238
Episode: 176: Time: 36.554813385009766 Total Reward: 591.8327402135142 Avg_Loss: 11.928119889327458
238
Episode: 177: Time: 36.35474443435669 Total Reward: 476.9557195571888 Avg_Loss: 7.055380394478806
238
Episode: 178: Time: 36.442588090896606 Total Reward: 172.55852842809855 Avg_Loss: 6.779840373692393
238
Episode: 179: Time: 35.30818247795105 Total Reward: 446.2541254125348 Avg_Loss: 6.731968079795356
238
Episode: 180: Time: 36.883068799972534 Total Reward: 199.6708463949871 Avg_Loss: 8.59946629179626
238
Episode: 181: Time: 34.887712240219116 Total Reward: 504.31506849314536 Avg_Loss: 9.84102786939685
238
Episode: 182: Time: 36.497398138046265 Total Reward: 482.31958762886165 Avg_Loss: 7.016598709491121
238
Episode: 183: Time: 36.04820013046265 Total Reward: 272.81609195402024 Avg_Loss: 7.130504953260181
238
Episode: 184: Time: 36.9753520488739 Total Reward: 435.8641975308603 Avg_Loss: 6.83872427008733
238
Episode: 185: Time: 35.40911841392517 Total Reward: 709.4280442804351 Avg_Loss: 7.9764529176118995
238
Episode: 186: Time: 36.13001012802124 Total Reward: 271.2420382165629 Avg_Loss: 7.734874608136025
238
Episode: 187: Time: 37.23241400718689 Total Reward: 389.8484848484777 Avg_Loss: 8.90450715016918
238
Episode: 188: Time: 35.378965616226196 Total Reward: 366.80555555554884 Avg_Loss: 7.8767743406175565
238
Episode: 189: Time: 36.41712546348572 Total Reward: 366.26760563379366 Avg_Loss: 7.678268083003389
238
Episode: 190: Time: 34.97772979736328 Total Reward: 399.5848375451241 Avg_Loss: 8.013877677817305
238
Episode: 191: Time: 38.73621463775635 Total Reward: 377.6027397260198 Avg_Loss: 6.934708036294504
238
Episode: 192: Time: 35.95357942581177 Total Reward: 483.6163522012474 Avg_Loss: 6.635903539276924
238
Episode: 193: Time: 36.57771015167236 Total Reward: 326.05263157894314 Avg_Loss: 7.650282979011536
238
Episode: 194: Time: 36.0867805480957 Total Reward: 239.36532507739716 Avg_Loss: 7.667929359844753
238
Episode: 195: Time: 36.6445746421814 Total Reward: 456.7241379310293 Avg_Loss: 7.608145234965477
238
Episode: 196: Time: 36.528414726257324 Total Reward: 353.90510948905046 Avg_Loss: 7.8183009899964855
238
Episode: 197: Time: 36.52208924293518 Total Reward: 124.3732193732229 Avg_Loss: 7.743408139012441
238
Episode: 198: Time: 38.52018594741821 Total Reward: 444.3258426966255 Avg_Loss: 7.69482678575676
238
Episode: 199: Time: 35.99688744544983 Total Reward: 255.31847133757282 Avg_Loss: 7.790470990313201
self.validation_rewards = [322.66424841618806, 492.1341337621951, -94.999999999999, 565.3297523653507]
Validation Mean Reward: 565.3297523653507 Validation Std Reward: 184.32969918722142
Test Mean Reward: 505.7495048573211 Test Std Reward: 178.09727575939715



Plots for HardUpdateDQN ($C = 100$)



Changing update_freq (C) to 1 and seeing how model works

```
In [ ]: import DQN
import utils
import torch

trainerHardUpdateDQN = DQN.HardUpdateDQN(EnvWrapper(env),
    model.Nature_Paper_Conv,
    update_freq = 1,
    lr = 0.00025,
    gamma = 0.95,
    buffer_size=20000,
    batch_size=16,
    loss_fn = "mse_loss",
    use_wandb = False,
    device = 'cpu',
    seed = 42,
    epsilon_scheduler = utils.exponential_decay(1, 1000, 0.1),
    save_path = utils.get_save_path("DoubleDQN_HardUpdates/", "./runs/"))

trainerHardUpdateDQN.train(200, 50, 30, 50, 50)

saving to ./runs/DoubleDQN_HardUpdates/run12
```



```
/content/drive/MyDrive/DQN_Project4/DQN.py:370: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
states = torch.tensor(states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:371: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
actions = torch.tensor(actions).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:372: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
rewards = torch.tensor(rewards).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:373: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
next_states = torch.tensor(next_states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/DQN_Project4/DQN.py:374: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
done = torch.tensor(done).clone().detach().float().to(self.device)
```

79

Episode: 0: Time: 20.721867322921753 Total Reward: -73.41726618705059 Avg_Loss: 0.46594999966364875

238

Episode: 1: Time: 32.38112187385559 Total Reward: -66.15384615384669 Avg_Loss: 0.4240172258460847

238

Episode: 2: Time: 27.80776286125183 Total Reward: -65.80291970802982 Avg_Loss: 0.4790104076563686

238

Episode: 3: Time: 31.430201768875122 Total Reward: -55.7142857142866 Avg_Loss: 0.454881727628942

238

Episode: 4: Time: 35.57332444190979 Total Reward: -27.65993265993332 Avg_Loss: 0.5237278950189342

238

Episode: 5: Time: 32.95370101928711 Total Reward: 0.07042253521030917 Avg_Loss: 0.8608980545961932

196

Episode: 6: Time: 24.21235418319702 Total Reward: -105.68091603053496 Avg_Loss: 0.7887474948787415

238

Episode: 7: Time: 32.46161651611328 Total Reward: -16.87500000000048 Avg_Loss: 8.342604024398351

182

Episode: 8: Time: 23.10585308074951 Total Reward: -56.657971014491814 Avg_Loss: 5.172080755366811

104

Episode: 9: Time: 14.171122789382935 Total Reward: -77.10606060606108 Avg_Loss: 3.4021668908759377

238

Episode: 10: Time: 34.04291224479675 Total Reward: -32.9084967320268 Avg_Loss: 6.744917198373866

238

Episode: 11: Time: 33.921199798583984 Total Reward: 6.083032490974334 Avg_Loss: 4.381361213567502

238

Episode: 12: Time: 33.633169412612915 Total Reward: 43.97280966767646 Avg_Loss: 3.8918215649969436

238

Episode: 13: Time: 33.194581031799316 Total Reward: -30.153583617748062 Avg_Loss: 2.2146116497010744

238

Episode: 14: Time: 32.20308709144592 Total Reward: -27.20338983050922 Avg_Loss: 2.4328860912565924

238

Episode: 15: Time: 32.68602514266968 Total Reward: -49.983922829582596 Avg_Loss: 1.614165282863028

238

Episode: 16: Time: 35.456440448760986 Total Reward: 34.8245614035113 Avg_Loss: 3.9020373678507925

238

Episode: 17: Time: 32.639907121658325 Total Reward: -27.20338983050892 Avg_Loss: 2.949668688464816

238

Episode: 18: Time: 32.86588263511658 Total Reward: 129.29906542056506 Avg_Loss: 3.2641427160686804

238

Episode: 19: Time: 34.20125865936279 Total Reward: 15.032362459546164 Avg_Loss: 2.5978905222370843

238

Episode: 20: Time: 32.969178438186646 Total Reward: 95.14084507042693 Avg_Loss: 1.8603577725902325

238

Episode: 21: Time: 33.06597304344177 Total Reward: 354.32432432432466 Avg_Loss: 4.498703270837539

238

Episode: 22: Time: 38.08862376213074 Total Reward: 53.936170212766356 Avg_Loss: 3.0550476672018276

238

Episode: 23: Time: 33.72225499153137 Total Reward: -16.146953405018706 Avg_Loss: 3.196063397311363

238

Episode: 24: Time: 34.27217245101929 Total Reward: 155.00000000000452 Avg_Loss: 3.533188750137802

238

Episode: 25: Time: 32.380510091781616 Total Reward: 102.80219780220153 Avg_Loss: 4.919975744826453

238

Episode: 26: Time: 32.66153931617737 Total Reward: 111.29370629370914 Avg_Loss: 2.7499527886447286

238

Episode: 27: Time: 36.249260663986206 Total Reward: 352.552447552444 Avg_Loss: 4.844248261271405

238

Episode: 28: Time: 33.42469358444214 Total Reward: 86.81818181818605 Avg_Loss: 3.7479116902882312

238

Episode: 29: Time: 33.15224099159241 Total Reward: 271.19718309859576 Avg_Loss: 4.100682057002011

238

Episode: 30: Time: 34.68745470046997 Total Reward: 621.8458781361899 Avg_Loss: 4.853933112836685

238

Episode: 31: Time: 32.34080672264099 Total Reward: 48.75000000000033 Avg_Loss: 5.932713002216916

238

Episode: 32: Time: 32.923243045806885 Total Reward: 144.54372623574523 Avg_Loss: 5.245924466422626

238

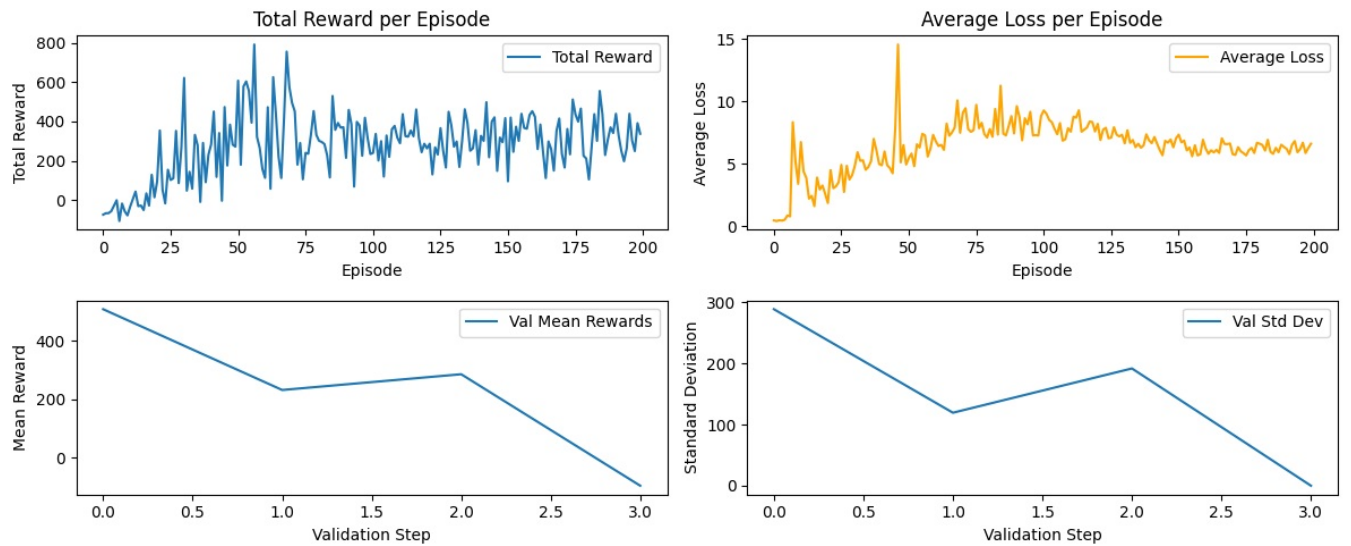
Episode: 33: Time: 35.57374691963196 Total Reward: 58.594771241834316 Avg_Loss: 5.284810659765196

238
Episode: 34: Time: 33.157814502716064 Total Reward: 332.2151898734083 Avg_Loss: 4.530098164908025
238
Episode: 35: Time: 34.52692747116089 Total Reward: 279.21383647798024 Avg_Loss: 4.744045081008382
238
Episode: 36: Time: 33.66609764099121 Total Reward: -8.621262458472012 Avg_Loss: 5.218954932539403
238
Episode: 37: Time: 33.237218141555786 Total Reward: 291.29283489096537 Avg_Loss: 7.003669247782531
238
Episode: 38: Time: 34.165151834487915 Total Reward: 91.56716417910695 Avg_Loss: 6.0738220978684785
238
Episode: 39: Time: 35.02862286567688 Total Reward: 230.94936708860885 Avg_Loss: 4.986533112636133
238
Episode: 40: Time: 35.867074966430664 Total Reward: 283.1818181818139 Avg_Loss: 4.884848899701062
238
Episode: 41: Time: 33.33823800086975 Total Reward: 450.8015267175507 Avg_Loss: 5.997513083349757
238
Episode: 42: Time: 33.68402338027954 Total Reward: 118.81578947368848 Avg_Loss: 4.884010349001203
238
Episode: 43: Time: 33.89546465873718 Total Reward: 341.36363636363717 Avg_Loss: 4.662604365278693
79
Episode: 44: Time: 11.467784643173218 Total Reward: -2.686440677965237 Avg_Loss: 4.247128472298006
238
Episode: 45: Time: 35.34974789619446 Total Reward: 473.4931506849211 Avg_Loss: 7.971088799107976
238
Episode: 46: Time: 34.32075524330139 Total Reward: 176.4285714285757 Avg_Loss: 14.558846316167287
238
Episode: 47: Time: 33.547449827194214 Total Reward: 384.2332268370542 Avg_Loss: 5.123024759923711
238
Episode: 48: Time: 32.99090886116028 Total Reward: 282.55102040816337 Avg_Loss: 6.504385204250071
238
Episode: 49: Time: 34.696030616760254 Total Reward: 272.3469387755114 Avg_Loss: 4.945918475129023
self.validation_rewards = [508.4672394716112]
Validation Mean Reward: 508.4672394716112 Validation Std Reward: 288.46116440478903
238
Episode: 50: Time: 33.52483010292053 Total Reward: 607.7027027026904 Avg_Loss: 5.469681401463116
238
Episode: 51: Time: 35.294859647750854 Total Reward: 180.22935779816987 Avg_Loss: 5.826138575788305
238
Episode: 52: Time: 33.56075954437256 Total Reward: 578.4693877550949 Avg_Loss: 4.8098399589041705
238
Episode: 53: Time: 33.36908173561096 Total Reward: 603.9247311827875 Avg_Loss: 6.567156544002164
238
Episode: 54: Time: 35.569392919540405 Total Reward: 555.7352941176364 Avg_Loss: 6.309279327633
238
Episode: 55: Time: 33.06035566329956 Total Reward: 394.2086330935141 Avg_Loss: 7.402850114497818
238
Episode: 56: Time: 33.041632413864136 Total Reward: 791.639676113346 Avg_Loss: 7.280218751240177
238
Episode: 57: Time: 34.31560683250427 Total Reward: 321.96113074203964 Avg_Loss: 5.587080704564808
238
Episode: 58: Time: 33.102314710617065 Total Reward: 266.0108303249128 Avg_Loss: 6.215921910370097
238
Episode: 59: Time: 35.75239109992981 Total Reward: 158.90625000000313 Avg_Loss: 7.319326724825787
238
Episode: 60: Time: 33.499486207962036 Total Reward: 113.79120879120873 Avg_Loss: 6.803701395748043
238
Episode: 61: Time: 33.678322076797485 Total Reward: 472.741935483868 Avg_Loss: 6.431671776566185
238
Episode: 62: Time: 34.952077865600586 Total Reward: 58.3333333333369 Avg_Loss: 6.519459272132201
238
Episode: 63: Time: 33.302629470825195 Total Reward: 625.8480565370943 Avg_Loss: 6.122511650834765
238
Episode: 64: Time: 33.51293754577637 Total Reward: 438.5570469798562 Avg_Loss: 8.221698782023262
238
Episode: 65: Time: 36.08828663825989 Total Reward: 221.4556962025341 Avg_Loss: 7.252658335601582
238
Episode: 66: Time: 34.11957550048828 Total Reward: 112.97720797721189 Avg_Loss: 7.515248142621097
238
Episode: 67: Time: 34.55330967903137 Total Reward: 410.0847457627014 Avg_Loss: 7.913725343071112
238
Episode: 68: Time: 32.97739791870117 Total Reward: 755.5338078291701 Avg_Loss: 10.08147442941906
238
Episode: 69: Time: 33.13682746887207 Total Reward: 574.3877551020278 Avg_Loss: 7.490592150127187
238
Episode: 70: Time: 35.9081711769104 Total Reward: 494.3536121672925 Avg_Loss: 9.092488059476644
238
Episode: 71: Time: 33.337061166763306 Total Reward: 452.101449275357 Avg_Loss: 9.455928359712873
238
Episode: 72: Time: 34.35108041763306 Total Reward: 180.86206896552028 Avg_Loss: 7.7859547308513095
238
Episode: 73: Time: 33.83609104156494 Total Reward: 290.852090032155 Avg_Loss: 7.565759786036836
238
Episode: 74: Time: 32.89837026596069 Total Reward: 106.38888888889355 Avg_Loss: 7.7066393805151225
238
Episode: 75: Time: 35.185325622558594 Total Reward: 240.64013840830535 Avg_Loss: 9.724403735970249
238
Episode: 76: Time: 35.18868947029114 Total Reward: 236.25000000000256 Avg_Loss: 7.850266878845311
238

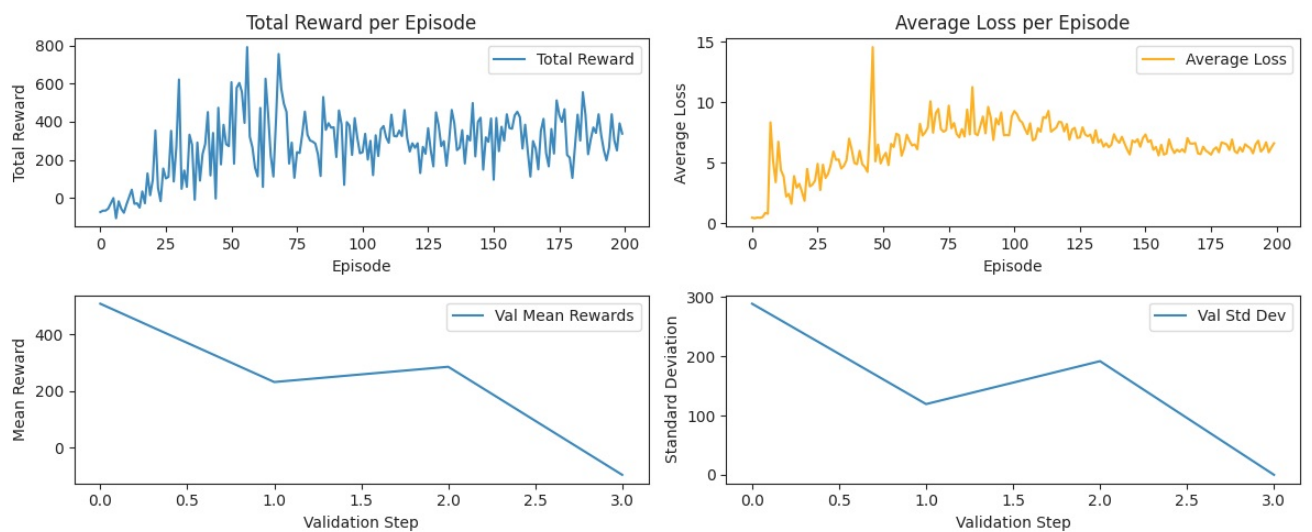
Episode: 77: Time: 35.28085279464722 Total Reward: 334.50819672131007 Avg_Loss: 8.28615482514646
238
Episode: 78: Time: 34.155696868896484 Total Reward: 453.2758620689583 Avg_Loss: 7.376645250856376
238
Episode: 79: Time: 34.3500337600708 Total Reward: 331.66666666666697 Avg_Loss: 7.108012139296331
238
Episode: 80: Time: 34.341594219207764 Total Reward: 301.9230769230777 Avg_Loss: 7.7887632591383795
238
Episode: 81: Time: 33.67378211021423 Total Reward: 296.0256410256401 Avg_Loss: 7.166092093251333
238
Episode: 82: Time: 36.54686975479126 Total Reward: 285.2816901408471 Avg_Loss: 9.396728688929262
238
Episode: 83: Time: 33.658730268478394 Total Reward: 228.33333333333746 Avg_Loss: 7.340653504393682
238
Episode: 84: Time: 34.712263107299805 Total Reward: 115.68249258160256 Avg_Loss: 11.259680014197565
238
Episode: 85: Time: 33.90567445755005 Total Reward: 529.9999999999895 Avg_Loss: 7.419785459502404
238
Episode: 86: Time: 33.67048478126526 Total Reward: 357.7687296416926 Avg_Loss: 7.279103902708583
238
Episode: 87: Time: 36.38998532295227 Total Reward: 392.7192982456071 Avg_Loss: 8.022496697782469
238
Episode: 88: Time: 33.6198525428772 Total Reward: 369.28571428571144 Avg_Loss: 8.828368627223648
238
Episode: 89: Time: 34.89674949645996 Total Reward: 371.16541353383104 Avg_Loss: 7.435796141123571
238
Episode: 90: Time: 33.4136118888855 Total Reward: 215.34482758621056 Avg_Loss: 9.621247151318718
238
Episode: 91: Time: 33.35166358947754 Total Reward: 459.2635658914623 Avg_Loss: 8.745097967756896
238
Episode: 92: Time: 34.06265115737915 Total Reward: 384.8534798534729 Avg_Loss: 6.888948519189818
238
Episode: 93: Time: 35.4205162525177 Total Reward: 69.38356164383534 Avg_Loss: 8.652834314759039
238
Episode: 94: Time: 34.70204305648804 Total Reward: 398.4210526315718 Avg_Loss: 8.163305841574148
238
Episode: 95: Time: 33.84406781196594 Total Reward: 380.177304964535 Avg_Loss: 9.166471653625745
238
Episode: 96: Time: 34.17030692100525 Total Reward: 225.98765432099134 Avg_Loss: 7.281014762505763
238
Episode: 97: Time: 34.44167113304138 Total Reward: 419.08450704225197 Avg_Loss: 7.301933194408898
238
Episode: 98: Time: 34.30744528770447 Total Reward: 315.8761329305125 Avg_Loss: 7.271081377979086
238
Episode: 99: Time: 36.66026282310486 Total Reward: 234.82456140351212 Avg_Loss: 8.840259007045201
self.validation_rewards = [508.4672394716112, 232.3444090615343]
Validation Mean Reward: 232.3444090615343 Validation Std Reward: 119.28534493788133
238
Episode: 100: Time: 33.96566843986511 Total Reward: 240.59322033898283 Avg_Loss: 9.275359413703951
238
Episode: 101: Time: 35.084922790527344 Total Reward: 337.05574912891893 Avg_Loss: 9.01319002804636
238
Episode: 102: Time: 33.79602336883545 Total Reward: 201.92832764505374 Avg_Loss: 8.519939102545505
238
Episode: 103: Time: 35.22800064086914 Total Reward: 300.27027027027145 Avg_Loss: 8.333175715278177
238
Episode: 104: Time: 34.526440143585205 Total Reward: 120.23178807947346 Avg_Loss: 7.804715875817948
238
Episode: 105: Time: 33.26515293121338 Total Reward: 329.0282685512296 Avg_Loss: 7.371365538665226
238
Episode: 106: Time: 35.41640329360962 Total Reward: 220.49295774648175 Avg_Loss: 8.265911696337852
238
Episode: 107: Time: 33.16652297973633 Total Reward: 360.88235294117595 Avg_Loss: 6.860111226053799
238
Episode: 108: Time: 34.66706109046936 Total Reward: 377.97297297296626 Avg_Loss: 6.970803023386402
238
Episode: 109: Time: 35.22807168960571 Total Reward: 318.19444444444264 Avg_Loss: 7.850494361725174
238
Episode: 110: Time: 34.83401012420654 Total Reward: 289.839650145775 Avg_Loss: 7.580959866026871
238
Episode: 111: Time: 33.855624198913574 Total Reward: 437.467532467529 Avg_Loss: 8.77362048726122
238
Episode: 112: Time: 33.97933506965637 Total Reward: 325.68965517240997 Avg_Loss: 8.699468874630808
238
Episode: 113: Time: 34.32369065284729 Total Reward: 323.1818181818112 Avg_Loss: 9.287175621806073
238
Episode: 114: Time: 34.43629431724548 Total Reward: 354.2307692307662 Avg_Loss: 7.575869617341947
238
Episode: 115: Time: 36.245128870010376 Total Reward: 325.19543973940483 Avg_Loss: 7.746356608987856
238
Episode: 116: Time: 33.72799324989319 Total Reward: 461.3636363636317 Avg_Loss: 7.913599245688495
238
Episode: 117: Time: 34.4362850189209 Total Reward: 315.42345276872595 Avg_Loss: 8.390657043757558
238
Episode: 118: Time: 33.67625284194946 Total Reward: 244.04109589041298 Avg_Loss: 8.349048755249056
238
Episode: 119: Time: 33.74807000160217 Total Reward: 287.14285714284944 Avg_Loss: 7.5953980669254015
238
Episode: 120: Time: 33.62247323989868 Total Reward: 263.55263157894933 Avg_Loss: 8.20648783695798

238
Episode: 121: Time: 37.573585987091064 Total Reward: 287.3529411764709 Avg_Loss: 6.904515290961546
238
Episode: 122: Time: 34.49150490760803 Total Reward: 131.1146496815329 Avg_Loss: 7.744733643631975
238
Episode: 123: Time: 35.37292742729187 Total Reward: 268.6363636363663 Avg_Loss: 7.885405636635147
238
Episode: 124: Time: 33.99133253097534 Total Reward: 232.04402515723584 Avg_Loss: 7.049236911184647
238
Episode: 125: Time: 34.898701667785645 Total Reward: 366.2794612794597 Avg_Loss: 7.105040249704313
238
Episode: 126: Time: 34.12392330169678 Total Reward: 250.5657492354792 Avg_Loss: 7.930824148554762
238
Episode: 127: Time: 36.297555446624756 Total Reward: 166.14649681529025 Avg_Loss: 7.335552611270873
238
Episode: 128: Time: 34.29846119880676 Total Reward: 449.55445544554027 Avg_Loss: 7.204642825767774
238
Episode: 129: Time: 34.414153814315796 Total Reward: 382.1929824561332 Avg_Loss: 7.33113659329775
238
Episode: 130: Time: 33.76852107048035 Total Reward: 272.4911660777411 Avg_Loss: 6.642136961472135
238
Episode: 131: Time: 35.268495321273804 Total Reward: 298.44262295082007 Avg_Loss: 7.692782233743107
238
Episode: 132: Time: 34.312016010284424 Total Reward: 169.3312101910856 Avg_Loss: 6.687735102757686
238
Episode: 133: Time: 36.373594760894775 Total Reward: 300.7597173144828 Avg_Loss: 6.910589342357731
238
Episode: 134: Time: 33.97632884979248 Total Reward: 462.0934256055321 Avg_Loss: 6.335641601506402
238
Episode: 135: Time: 34.91113567352295 Total Reward: 391.8913857677883 Avg_Loss: 6.610745024280388
238
Episode: 136: Time: 33.620415449142456 Total Reward: 251.4566929133895 Avg_Loss: 6.293264729636056
238
Episode: 137: Time: 34.773966550827026 Total Reward: 263.5657370517912 Avg_Loss: 6.4858329917202475
238
Episode: 138: Time: 34.63047170639038 Total Reward: 356.2195121951161 Avg_Loss: 7.364915853788872
238
Episode: 139: Time: 37.16510581970215 Total Reward: 181.07361963190525 Avg_Loss: 6.8896010743469756
238
Episode: 140: Time: 34.959299087524414 Total Reward: 327.22222222221546 Avg_Loss: 6.657001656644485
238
Episode: 141: Time: 34.57163166999817 Total Reward: 301.551724137925 Avg_Loss: 7.151743312843707
238
Episode: 142: Time: 34.38962435722351 Total Reward: 498.3333333333256 Avg_Loss: 6.609022906848362
238
Episode: 143: Time: 36.0903754234314 Total Reward: 218.432835820899 Avg_Loss: 6.0966441310754345
238
Episode: 144: Time: 34.01034140586853 Total Reward: 399.91525423728274 Avg_Loss: 5.6881451992427605
238
Episode: 145: Time: 36.304813861846924 Total Reward: 421.3934426229385 Avg_Loss: 6.844693319136355
238
Episode: 146: Time: 35.20285129547119 Total Reward: 149.11764705882695 Avg_Loss: 6.696632319638709
238
Episode: 147: Time: 34.546635389328 Total Reward: 318.4615384615344 Avg_Loss: 6.95123540653902
238
Episode: 148: Time: 34.88696098327637 Total Reward: 294.99999999999847 Avg_Loss: 6.325359194719491
238
Episode: 149: Time: 34.61779594421387 Total Reward: 417.3674911660747 Avg_Loss: 7.044084952658966
self.validation_rewards = [508.4672394716112, 232.3444090615343, 286.154784299767]
Validation Mean Reward: 286.154784299767 Validation Std Reward: 191.65451944246337
238
Episode: 150: Time: 34.074594497680664 Total Reward: 96.12627986348363 Avg_Loss: 7.342735391704976
238
Episode: 151: Time: 36.27798295021057 Total Reward: 420.15151515151246 Avg_Loss: 6.7231986587788874
238
Episode: 152: Time: 35.729146003723145 Total Reward: 246.10787172011922 Avg_Loss: 6.867004828793662
238
Episode: 153: Time: 34.06032085418701 Total Reward: 373.86446886446254 Avg_Loss: 6.076343624531722
238
Episode: 154: Time: 35.111958503723145 Total Reward: 301.55172413793036 Avg_Loss: 6.309920321993467
238
Episode: 155: Time: 34.37384867668152 Total Reward: 439.24657534245966 Avg_Loss: 5.607893066746848
238
Episode: 156: Time: 34.827202558517456 Total Reward: 367.68656716417183 Avg_Loss: 6.490620293537108
238
Episode: 157: Time: 34.645328521728516 Total Reward: 363.4615384615348 Avg_Loss: 5.676618533975938
238
Episode: 158: Time: 36.583539962768555 Total Reward: 433.36879432623516 Avg_Loss: 5.774372451946515
238
Episode: 159: Time: 34.506606578826904 Total Reward: 452.79411764705344 Avg_Loss: 6.934149530755372
238
Episode: 160: Time: 36.21381211280823 Total Reward: 424.03114186850377 Avg_Loss: 6.222259251510396
238
Episode: 161: Time: 35.007418632507324 Total Reward: 259.7297297297313 Avg_Loss: 5.807952648952227
238
Episode: 162: Time: 36.22668266296387 Total Reward: 384.72972972972576 Avg_Loss: 6.08836660114657
238
Episode: 163: Time: 34.49221134185791 Total Reward: 245.00000000000443 Avg_Loss: 5.918276170722577
238

Episode: 164: Time: 37.307546854019165 Total Reward: 112.61245674740891 Avg_Loss: 6.117974982041271
238
Episode: 165: Time: 35.813599824905396 Total Reward: 298.8461538461505 Avg_Loss: 5.878651450662052
238
Episode: 166: Time: 35.48588013648987 Total Reward: 260.14018691589166 Avg_Loss: 7.042895257973871
238
Episode: 167: Time: 35.275935649871826 Total Reward: 150.7337883959088 Avg_Loss: 6.5942476927733225
238
Episode: 168: Time: 34.84596514701843 Total Reward: 353.16053511705684 Avg_Loss: 6.582910555751384
238
Episode: 169: Time: 35.60464310646057 Total Reward: 415.2739726027363 Avg_Loss: 6.6309113883170765
238
Episode: 170: Time: 35.648611068725586 Total Reward: 234.26829268293125 Avg_Loss: 5.778484181696627
238
Episode: 171: Time: 36.03713297843933 Total Reward: 166.12759643917315 Avg_Loss: 5.715544024936292
238
Episode: 172: Time: 34.970871686935425 Total Reward: 362.14285714285245 Avg_Loss: 6.335686828909802
238
Episode: 173: Time: 36.19947147369385 Total Reward: 233.4132841328428 Avg_Loss: 6.008310669610481
238
Episode: 174: Time: 34.65247392654419 Total Reward: 512.1428571428531 Avg_Loss: 5.85541345892834
238
Episode: 175: Time: 36.140573263168335 Total Reward: 436.83520599250573 Avg_Loss: 5.674165680628865
238
Episode: 176: Time: 36.362457513809204 Total Reward: 399.6619217081802 Avg_Loss: 6.1033735871315
238
Episode: 177: Time: 35.97537589073181 Total Reward: 465.8856088560831 Avg_Loss: 6.280978455263026
238
Episode: 178: Time: 35.47777342796326 Total Reward: 226.07023411371722 Avg_Loss: 5.871358988665733
238
Episode: 179: Time: 35.731717348098755 Total Reward: 211.9306930693082 Avg_Loss: 6.690611206683792
238
Episode: 180: Time: 36.640477895736694 Total Reward: 105.62695924765251 Avg_Loss: 6.610781052533318
238
Episode: 181: Time: 34.842846632003784 Total Reward: 271.4383561643839 Avg_Loss: 6.517193408573375
238
Episode: 182: Time: 37.72253441810608 Total Reward: 437.6460481099573 Avg_Loss: 6.059552439621517
238
Episode: 183: Time: 36.67394399642944 Total Reward: 301.5517241379298 Avg_Loss: 6.9280908253012585
238
Episode: 184: Time: 35.03252983093262 Total Reward: 555.2057613168652 Avg_Loss: 5.968892071427417
238
Episode: 185: Time: 36.207966566085815 Total Reward: 436.3653136531319 Avg_Loss: 5.802988736569381
238
Episode: 186: Time: 35.80506229400635 Total Reward: 229.8407643312137 Avg_Loss: 6.2804800998263
238
Episode: 187: Time: 37.25336170196533 Total Reward: 308.03030303029817 Avg_Loss: 5.930805919551048
238
Episode: 188: Time: 37.70724177360535 Total Reward: 370.2777777777773 Avg_Loss: 6.53351728605623
238
Episode: 189: Time: 35.49971103668213 Total Reward: 341.619718309856 Avg_Loss: 6.352570747627931
238
Episode: 190: Time: 36.66112971305847 Total Reward: 439.29602888085833 Avg_Loss: 6.198747231178925
238
Episode: 191: Time: 35.7035117149353 Total Reward: 329.65753424657555 Avg_Loss: 5.7829307738472435
238
Episode: 192: Time: 37.02051067352295 Total Reward: 250.91194968553853 Avg_Loss: 6.522680977813336
238
Episode: 193: Time: 36.656330585479736 Total Reward: 198.23308270677134 Avg_Loss: 6.825963059894177
238
Episode: 194: Time: 37.52506113052368 Total Reward: 264.1331269349888 Avg_Loss: 5.929789446482138
238
Episode: 195: Time: 36.85956406593323 Total Reward: 439.4827586206812 Avg_Loss: 6.1778782931696465
238
Episode: 196: Time: 37.083842277526855 Total Reward: 302.81021897810444 Avg_Loss: 6.693987593430431
238
Episode: 197: Time: 37.02628684043884 Total Reward: 249.72934472934915 Avg_Loss: 5.870216970183268
238
Episode: 198: Time: 37.28309726715088 Total Reward: 391.8913857677816 Avg_Loss: 6.265467955785639
238
Episode: 199: Time: 37.64571475982666 Total Reward: 338.12101910827636 Avg_Loss: 6.616060708250318
self.validation_rewards = [508.4672394716112, 232.3444090615343, 286.154784299767, -94.99999999999903]
Validation Mean Reward: -94.99999999999903 Validation Std Reward: 7.406932321938053e-14
Test Mean Reward: 561.0064446524763 Test Std Reward: 257.9963835482962

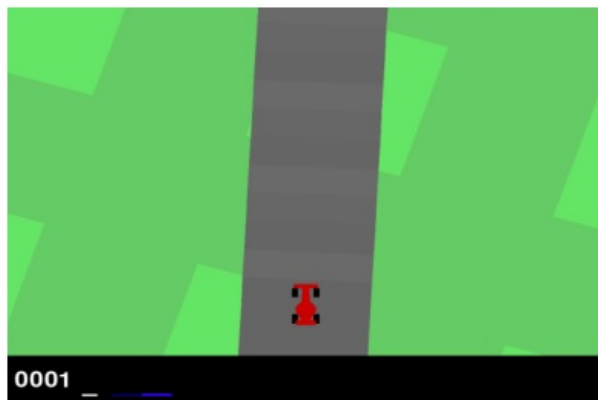
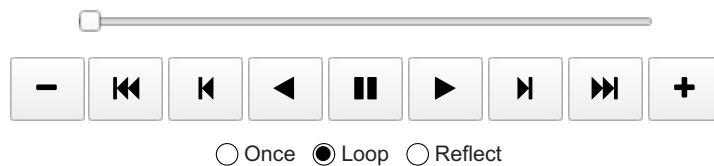


Plots for HardUpdatedQNN (C=1)



```
In [ ]: total_rewards, frames = trainerHardUpdatedQNN.play_episode(0, True, 42)
anim = animate(frames)
HTML(anim.to_jshtml())
```

Out[]:



Soft Updates (5 points)

A more recent improvement is to use soft updates, also known as Polyak averaging, where the target network is updated with a small fraction of the current model weights every step. In other words:

$$\theta_{target} = \tau \theta_{model} + (1 - \tau) \theta_{target}$$

for some $\tau \ll 1$
 $\ll 1$

Please implement this by implementing the `_update_model` class in `SoftUpdateDQN` in `DQN.py`.

```
In [ ]: import DQN
import utils
import torch

traineSoftUpdatedQN = DQN.SoftUpdateDQN(EnvWrapper(env),
    model.Nature_Paper_Conv,
    tau = 0.01,
    update_freq = 1,
    lr = 0.00025,
    gamma = 0.95,
    buffer_size=20000,
    batch_size=16,
    loss_fn = "mse_loss",
    use_wandb = False,
    device = 'cuda',
    seed = 42,
    epsilon_scheduler = utils.exponential_decay(1, 1000, 0.1),
    save_path = utils.get_save_path("DoubleDQN_SoftUpdates", "./runs/"))
```

```
traineSoftUpdatedDQN.train(200,50,30,50,50)
```

```
saving to ./runs/DoubleDQN_SoftUpdates/run1
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:370: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
states = torch.tensor(states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:371: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
actions = torch.tensor(actions).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:372: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
rewards = torch.tensor(rewards).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:373: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
next_states = torch.tensor(next_states).clone().detach().to(self.device)
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:374: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
dones = torch.tensor(dones).clone().detach().float().to(self.device)
```

```
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/conv.py:456: UserWarning: Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalizeDescriptor Failed cudnn_status: CUDNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
```

```
return F.conv2d(input, weight, bias, self.stride,
```

```
79
```

```
Episode: 0: Time: 13.326700210571289 Total Reward: 23.705035971223108 Avg_Loss: 1.469689637705495
```

```
238
```

```
Episode: 1: Time: 13.567751169204712 Total Reward: -62.948717948718524 Avg_Loss: 1.321579295202714
```

```
238
```

```
Episode: 2: Time: 13.701852083206177 Total Reward: -76.7518248175183 Avg_Loss: 0.8969183426189917
```

```
238
```

```
Episode: 3: Time: 13.44012999534607 Total Reward: -19.9999999999983 Avg_Loss: 0.8198925533464977
```

```
238
```

```
Episode: 4: Time: 14.391453266143799 Total Reward: -47.86195286195345 Avg_Loss: 0.8398673954662406
```

```
238
```

```
Episode: 5: Time: 13.606728792190552 Total Reward: -66.83098591549364 Avg_Loss: 0.7368486327385264
```

```
238
```

```
Episode: 6: Time: 13.502348184585571 Total Reward: -60.64885496183301 Avg_Loss: 0.7523107328726089
```

```
238
```

```
Episode: 7: Time: 12.658262252807617 Total Reward: -75.4687500000001 Avg_Loss: 0.6846605717838437
```

```
238
```

```
Episode: 8: Time: 13.104485034942627 Total Reward: -4.4202898550725545 Avg_Loss: 0.7137227858565435
```

```
238
```

```
Episode: 9: Time: 13.543787479400635 Total Reward: -0.3030303030309591 Avg_Loss: 0.763532270098246
```

```
238
```

```
Episode: 10: Time: 14.656925439834595 Total Reward: -55.784313725490904 Avg_Loss: 0.8297896814696929
```

```
238
```

```
Episode: 11: Time: 12.89335823059082 Total Reward: 20.523465703972725 Avg_Loss: 0.830116284510293
```

```
238
```

```
Episode: 12: Time: 13.966840982437134 Total Reward: -22.492447129910065 Avg_Loss: 1.00099992706683
```

```
238
```

```
Episode: 13: Time: 12.993544340133667 Total Reward: -77.93515358361773 Avg_Loss: 0.8647236815583306
```

```
238
```

```
Episode: 14: Time: 12.03754472732544 Total Reward: -30.5932203389838 Avg_Loss: 1.0180730378615255
```

```
238
```

```
Episode: 15: Time: 13.128041982650757 Total Reward: -33.906752411576264 Avg_Loss: 0.914254657329381
```

```
238
```

```
Episode: 16: Time: 13.095637559890747 Total Reward: -59.91228070175508 Avg_Loss: 0.8748884927324888
```

```
238
```

```
Episode: 17: Time: 13.20998501777649 Total Reward: 121.94915254237452 Avg_Loss: 1.2043967987058544
```

```
150
```

```
Episode: 18: Time: 7.652414798736572 Total Reward: -106.84049844236796 Avg_Loss: 1.304408836911122
```

```
238
```

```
Episode: 19: Time: 13.458979845046997 Total Reward: 11.796116504855043 Avg_Loss: 4.1343551302219135
```

```
238
```

```
Episode: 20: Time: 12.949406623840332 Total Reward: 21.197183098593243 Avg_Loss: 4.302742837544749
```

```
238
```

```
Episode: 21: Time: 14.549209594726562 Total Reward: 127.97297297297723 Avg_Loss: 6.859713822994538
```

```
238
```

```
Episode: 22: Time: 12.836580753326416 Total Reward: -20.53191489361765 Avg_Loss: 1.5525186215096913
```

```
238
```

```
Episode: 23: Time: 13.51360297203064 Total Reward: 385.2867383512449 Avg_Loss: 4.528028569082503
```

```
238
```

```
Episode: 24: Time: 13.040712833404541 Total Reward: -24.48717948718019 Avg_Loss: 2.1807074426602915
```

```
238
```

```
Episode: 25: Time: 13.228745222091675 Total Reward: 33.20512820512719 Avg_Loss: 2.142090419761273
```

```
238
```

```
Episode: 26: Time: 13.144152641296387 Total Reward: 216.18881118881552 Avg_Loss: 4.938998523410879
```

```
238
```

```
Episode: 27: Time: 13.7564697265625 Total Reward: 307.0979020978971 Avg_Loss: 5.294402075414898
```

```
238
```

```
Episode: 28: Time: 13.760339736938477 Total Reward: 437.91536050156213 Avg_Loss: 3.1788207924916962
```

```
238
```

```
Episode: 29: Time: 13.315077304840088 Total Reward: 81.05633802816962 Avg_Loss: 3.3992783075996806
```

```
238
```

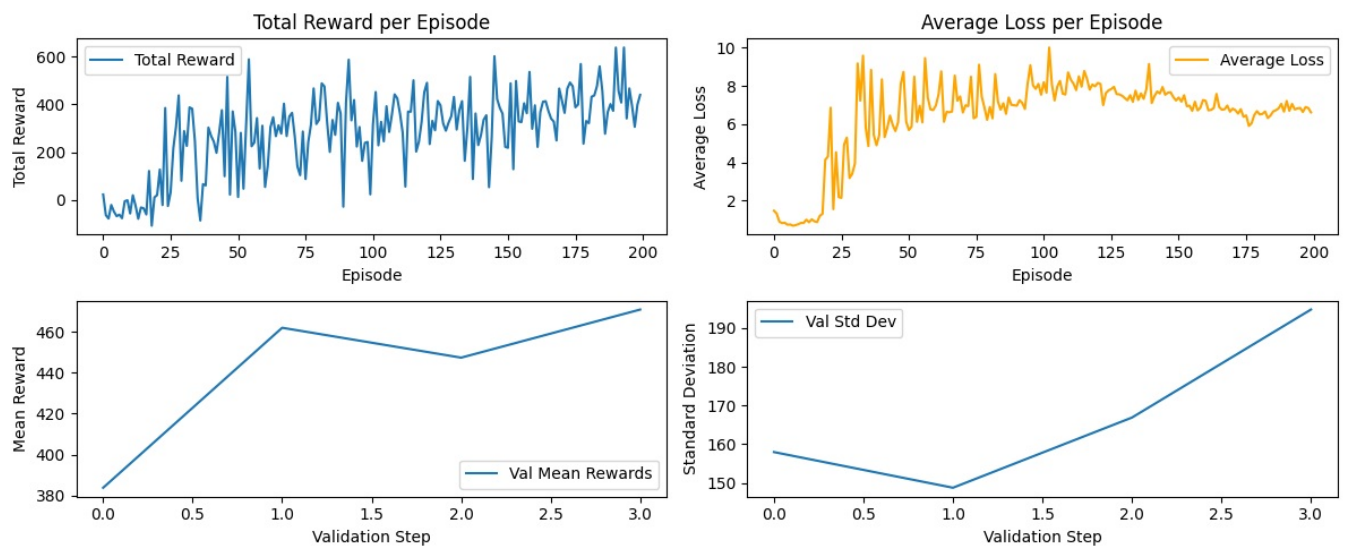
```
Episode: 30: Time: 13.369298219680786 Total Reward: 288.512544802855 Avg_Loss: 3.938761125029135
```


238
Episode: 31: Time: 13.70830512046814 Total Reward: 226.875000000004 Avg_Loss: 9.176033018329063
238
Episode: 32: Time: 15.133700370788574 Total Reward: 387.8897338403013 Avg_Loss: 7.228740025656063
238
Episode: 33: Time: 13.739211797714233 Total Reward: 382.1241830065301 Avg_Loss: 9.584527033216814
238
Episode: 34: Time: 13.804529905319214 Total Reward: 246.7721518987388 Avg_Loss: 5.819099605333905
238
Episode: 35: Time: 13.767117261886597 Total Reward: 8.773584905660666 Avg_Loss: 4.850127749082421
238
Episode: 36: Time: 13.66928744316101 Total Reward: -85.03322259136165 Avg_Loss: 8.838396502392632
238
Episode: 37: Time: 13.644853353500366 Total Reward: 66.99376947040615 Avg_Loss: 5.427894023630549
238
Episode: 38: Time: 13.163960218429565 Total Reward: 61.71641791044992 Avg_Loss: 4.895166409867151
238
Episode: 39: Time: 14.16471242904663 Total Reward: 303.7341772151834 Avg_Loss: 5.416563086900391
238
Episode: 40: Time: 14.101386785507202 Total Reward: 268.63636363635806 Avg_Loss: 8.342602211637658
238
Episode: 41: Time: 13.402343511581421 Total Reward: 244.6946564885523 Avg_Loss: 5.317994866801911
238
Episode: 42: Time: 13.81015157699585 Total Reward: 197.7631578947371 Avg_Loss: 5.8097209522203235
238
Episode: 43: Time: 15.02345609664917 Total Reward: 283.1818181818146 Avg_Loss: 6.447748450421486
238
Episode: 44: Time: 13.656998872756958 Total Reward: 376.18644067796225 Avg_Loss: 5.955969092725706
238
Episode: 45: Time: 14.009899139404297 Total Reward: 100.20547945205821 Avg_Loss: 5.638338160364568
238
Episode: 46: Time: 13.43434190750122 Total Reward: 515.7142857142782 Avg_Loss: 6.0995553468956665
238
Episode: 47: Time: 13.565205574035645 Total Reward: 23.210862619807386 Avg_Loss: 8.104060562969256
238
Episode: 48: Time: 13.459023237228394 Total Reward: 370.98639455782495 Avg_Loss: 8.734337597334084
238
Episode: 49: Time: 13.4054856300354 Total Reward: 285.9523809523738 Avg_Loss: 6.13191998280397
self.validation_rewards = [383.7412734227116]
Validation Mean Reward: 383.7412734227116 Validation Std Reward: 157.98158034239933
238
Episode: 50: Time: 14.082266569137573 Total Reward: 13.108108108107555 Avg_Loss: 5.692277298254125
238
Episode: 51: Time: 14.028562307357788 Total Reward: 281.14678899082367 Avg_Loss: 5.870765039650332
238
Episode: 52: Time: 15.616179466247559 Total Reward: 47.85714285714627 Avg_Loss: 8.480076283967795
238
Episode: 53: Time: 13.320319890975952 Total Reward: 313.6021505376351 Avg_Loss: 6.120349338575571
238
Episode: 54: Time: 13.623949527740479 Total Reward: 588.8235294117536 Avg_Loss: 6.96099716424942
238
Episode: 55: Time: 13.521708726882935 Total Reward: 225.14388489209074 Avg_Loss: 6.092929465430124
238
Episode: 56: Time: 13.460197687149048 Total Reward: 241.0323886639646 Avg_Loss: 9.45516841389051
238
Episode: 57: Time: 13.526642799377441 Total Reward: 343.16254416960913 Avg_Loss: 7.385650473983348
238
Episode: 58: Time: 14.006983995437622 Total Reward: 132.43682310469586 Avg_Loss: 6.770435459974434
238
Episode: 59: Time: 13.557442426681519 Total Reward: 311.2499999999985 Avg_Loss: 6.734587437465411
238
Episode: 60: Time: 13.274280071258545 Total Reward: 55.18315018315412 Avg_Loss: 6.977362984869661
238
Episode: 61: Time: 13.916624307632446 Total Reward: 143.70967741935908 Avg_Loss: 7.553704572575433
238
Episode: 62: Time: 13.765300989151001 Total Reward: 304.9999999999985 Avg_Loss: 8.759157662131205
238
Episode: 63: Time: 15.01680588722229 Total Reward: 346.6961130742015 Avg_Loss: 6.124974011623559
238
Episode: 64: Time: 13.428545713424683 Total Reward: 267.41610738255395 Avg_Loss: 6.64536050563099
238
Episode: 65: Time: 13.810390949249268 Total Reward: 313.22784810126393 Avg_Loss: 6.6183057327230435
238
Episode: 66: Time: 14.17553448677063 Total Reward: 278.21937321937304 Avg_Loss: 6.662349297719843
238
Episode: 67: Time: 13.6561861038208 Total Reward: 403.305084745757 Avg_Loss: 8.540165651746157
238
Episode: 68: Time: 13.451449871063232 Total Reward: 267.989323843416 Avg_Loss: 7.226174544887383
238
Episode: 69: Time: 13.169504404067993 Total Reward: 349.8979591836657 Avg_Loss: 7.447493314492602
238
Episode: 70: Time: 13.857493162155151 Total Reward: 365.0760456273729 Avg_Loss: 6.6064651358027415
238
Episode: 71: Time: 13.038028478622437 Total Reward: 267.31884057970467 Avg_Loss: 6.9863863241772695
238
Episode: 72: Time: 13.820097923278809 Total Reward: 140.10971786834074 Avg_Loss: 6.940815828427547
238
Episode: 73: Time: 13.768521547317505 Total Reward: 104.35691318328142 Avg_Loss: 8.480924905849104
238

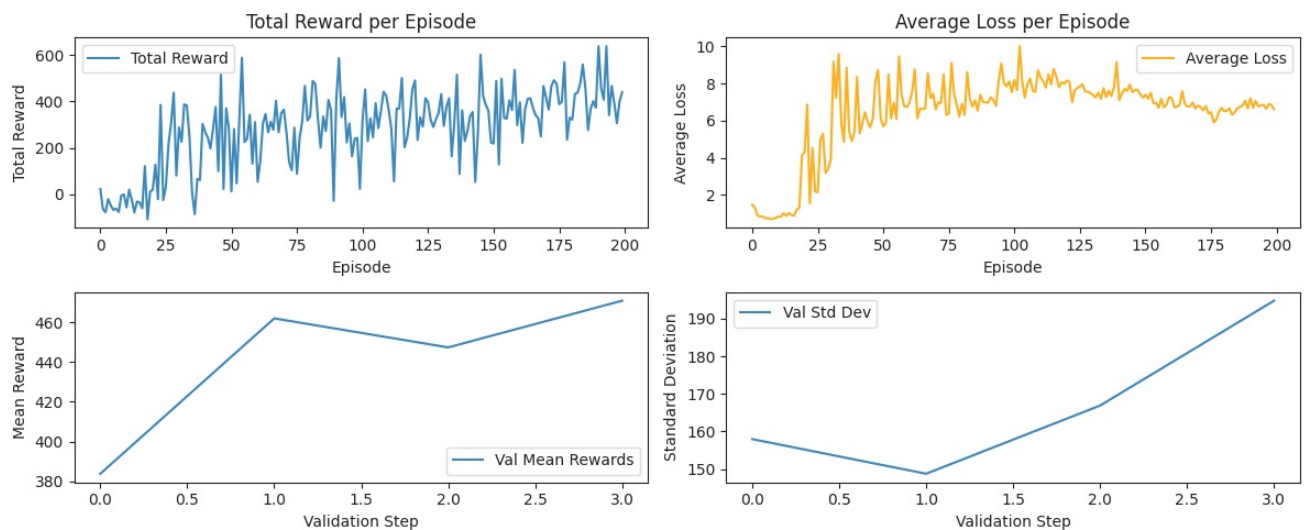
Episode: 74: Time: 13.747361183166504 Total Reward: 286.94444444444713 Avg_Loss: 6.308246000474241
238
Episode: 75: Time: 14.69457221031189 Total Reward: 88.3910034602113 Avg_Loss: 6.382353237697056
238
Episode: 76: Time: 13.919961929321289 Total Reward: 242.50000000000398 Avg_Loss: 9.10941597293405
238
Episode: 77: Time: 13.856954336166382 Total Reward: 314.8360655737683 Avg_Loss: 7.443801444869082
238
Episode: 78: Time: 13.709398984909058 Total Reward: 467.06896551723355 Avg_Loss: 6.869143282165046
238
Episode: 79: Time: 13.851748943328857 Total Reward: 318.3333333333317 Avg_Loss: 6.2247315179400085
238
Episode: 80: Time: 14.018870115280151 Total Reward: 335.7692307692279 Avg_Loss: 6.898883062250474
238
Episode: 81: Time: 14.44108271598816 Total Reward: 488.3333333333278 Avg_Loss: 6.30256762374349
238
Episode: 82: Time: 14.164555311203003 Total Reward: 475.4225352112575 Avg_Loss: 8.61781417472022
238
Episode: 83: Time: 13.804569005966187 Total Reward: 325.00000000000125 Avg_Loss: 7.180202271758008
238
Episode: 84: Time: 14.121054410934448 Total Reward: 201.7359050445137 Avg_Loss: 6.760353002728534
238
Episode: 85: Time: 15.028468370437622 Total Reward: 335.1470588235188 Avg_Loss: 7.091388855160785
238
Episode: 86: Time: 13.95597791671753 Total Reward: 273.0781758957692 Avg_Loss: 6.537323619137291
238
Episode: 87: Time: 13.927077054977417 Total Reward: 406.75438596490756 Avg_Loss: 7.392941864598699
238
Episode: 88: Time: 14.22401475906372 Total Reward: 362.79220779220066 Avg_Loss: 7.00966082050019
238
Episode: 89: Time: 13.544833660125732 Total Reward: -27.330827067669933 Avg_Loss: 6.99132609818162
238
Episode: 90: Time: 14.018372535705566 Total Reward: 409.7021943573636 Avg_Loss: 6.966927709198799
238
Episode: 91: Time: 13.566475629806519 Total Reward: 587.1705426356508 Avg_Loss: 7.254364929279359
238
Episode: 92: Time: 13.54275894165039 Total Reward: 333.5714285714213 Avg_Loss: 7.107937593420012
238
Episode: 93: Time: 13.644760847091675 Total Reward: 418.6986301369842 Avg_Loss: 6.7945040990324586
238
Episode: 94: Time: 13.931952953338623 Total Reward: 224.07894736842462 Avg_Loss: 8.120161386097179
238
Episode: 95: Time: 13.867369174957275 Total Reward: 305.7092198581524 Avg_Loss: 9.080810338008304
238
Episode: 96: Time: 14.079440116882324 Total Reward: 164.25925925926327 Avg_Loss: 8.02830038401259
238
Episode: 97: Time: 15.423447847366333 Total Reward: 239.5070422535257 Avg_Loss: 7.848358684728126
238
Episode: 98: Time: 14.137080430984497 Total Reward: 243.36858006042647 Avg_Loss: 8.108987776672139
238
Episode: 99: Time: 13.7586190700531 Total Reward: 24.298245614034414 Avg_Loss: 7.536820804371553
self.validation_rewards = [383.7412734227116, 462.0183116500987]
Validation Mean Reward: 462.0183116500987 Validation Std Reward: 148.78102976199796
238
Episode: 100: Time: 14.103368520736694 Total Reward: 318.5593220338976 Avg_Loss: 8.194449210868163
238
Episode: 101: Time: 13.923694133758545 Total Reward: 452.0383275261263 Avg_Loss: 7.648180955097455
238
Episode: 102: Time: 13.961272954940796 Total Reward: 229.23208191126116 Avg_Loss: 10.008925455958904
238
Episode: 103: Time: 13.7979416847229 Total Reward: 327.2972972972947 Avg_Loss: 8.07689225373148
238
Episode: 104: Time: 13.885501861572266 Total Reward: 246.05960264900966 Avg_Loss: 7.239785690267547
238
Episode: 105: Time: 13.951405763626099 Total Reward: 392.63250883391845 Avg_Loss: 7.97073281412365
238
Episode: 106: Time: 17.018390655517578 Total Reward: 285.28169014084233 Avg_Loss: 8.26170191945148
238
Episode: 107: Time: 13.368745803833008 Total Reward: 357.2058823529337 Avg_Loss: 7.595899287392111
238
Episode: 108: Time: 13.782185077667236 Total Reward: 442.16216216215736 Avg_Loss: 7.548776037552777
238
Episode: 109: Time: 13.749344825744629 Total Reward: 425.8333333333261 Avg_Loss: 8.707850932073192
238
Episode: 110: Time: 14.33589792251587 Total Reward: 362.72594752186313 Avg_Loss: 8.32935433227475
238
Episode: 111: Time: 13.905904054641724 Total Reward: 281.6233766233773 Avg_Loss: 8.115047484385867
238
Episode: 112: Time: 13.863234758377075 Total Reward: 56.724137931038584 Avg_Loss: 7.780953759906673
238
Episode: 113: Time: 13.756951808929443 Total Reward: 370.45454545454027 Avg_Loss: 8.501526681815877
238
Episode: 114: Time: 14.537854194641113 Total Reward: 369.6153846153806 Avg_Loss: 7.957917320628126
238
Episode: 115: Time: 14.409210920333862 Total Reward: 501.09120521171843 Avg_Loss: 8.780730071188021
238
Episode: 116: Time: 13.985441207885742 Total Reward: 203.18181818182174 Avg_Loss: 8.369362660816737
238
Episode: 117: Time: 15.988774299621582 Total Reward: 243.76221498370785 Avg_Loss: 7.805229351300151

238
Episode: 118: Time: 13.95806074142456 Total Reward: 326.23287671231935 Avg_Loss: 8.071287343482009
238
Episode: 119: Time: 13.539761066436768 Total Reward: 451.4285714285621 Avg_Loss: 7.979743682536759
238
Episode: 120: Time: 14.048824787139893 Total Reward: 490.5263157894657 Avg_Loss: 8.154869728729505
238
Episode: 121: Time: 14.343110799789429 Total Reward: 234.41176470588528 Avg_Loss: 8.101450418223854
238
Episode: 122: Time: 14.376106262207031 Total Reward: 331.75159235668156 Avg_Loss: 6.996688693511386
238
Episode: 123: Time: 14.227306365966797 Total Reward: 292.20538720538116 Avg_Loss: 7.621138009704461
238
Episode: 124: Time: 14.111971139907837 Total Reward: 414.4339622641471 Avg_Loss: 7.768354822607601
238
Episode: 125: Time: 13.814330101013184 Total Reward: 396.5824915824827 Avg_Loss: 7.839679331338706
238
Episode: 126: Time: 14.1208016872406 Total Reward: 320.90214067278225 Avg_Loss: 7.951482166763113
238
Episode: 127: Time: 14.12688398361206 Total Reward: 290.35031847133 Avg_Loss: 7.570606374940953
238
Episode: 128: Time: 16.20631694793701 Total Reward: 324.14191419141537 Avg_Loss: 7.5565080101750475
238
Episode: 129: Time: 13.53292727470398 Total Reward: 350.61403508770604 Avg_Loss: 7.487154477784614
238
Episode: 130: Time: 13.65527868270874 Total Reward: 431.50176678444785 Avg_Loss: 7.367711979801915
238
Episode: 131: Time: 13.822831869125366 Total Reward: 295.1639344262303 Avg_Loss: 7.247390553229997
238
Episode: 132: Time: 14.127495288848877 Total Reward: 376.3375796178298 Avg_Loss: 7.514168434784192
238
Episode: 133: Time: 13.559150218963623 Total Reward: 413.83392226147964 Avg_Loss: 7.16757561479296
238
Episode: 134: Time: 13.73190951347351 Total Reward: 164.51557093425933 Avg_Loss: 7.752257718258545
238
Episode: 135: Time: 13.47264814376831 Total Reward: 305.7490636704101 Avg_Loss: 7.265339944042077
238
Episode: 136: Time: 13.736228227615356 Total Reward: 515.2362204724325 Avg_Loss: 7.606010394937852
238
Episode: 137: Time: 13.386142015457153 Total Reward: 88.26693227091914 Avg_Loss: 7.321825790805977
238
Episode: 138: Time: 14.247496128082275 Total Reward: 362.31707317072795 Avg_Loss: 7.826897808483669
238
Episode: 139: Time: 15.787142753601074 Total Reward: 230.15337423313264 Avg_Loss: 9.146370768547058
238
Episode: 140: Time: 14.578141927719116 Total Reward: 271.6666666666697 Avg_Loss: 7.101855846513219
238
Episode: 141: Time: 14.038384914398193 Total Reward: 336.0344827586151 Avg_Loss: 7.465646741770897
238
Episode: 142: Time: 14.331217050552368 Total Reward: 354.99999999999653 Avg_Loss: 7.706446740807605
238
Episode: 143: Time: 13.656280755996704 Total Reward: 54.2537313432873 Avg_Loss: 7.603563850667296
238
Episode: 144: Time: 13.84778356552124 Total Reward: 264.3220338983089 Avg_Loss: 7.942401150695416
238
Episode: 145: Time: 13.502150774002075 Total Reward: 601.7213114753963 Avg_Loss: 7.536826034553912
238
Episode: 146: Time: 14.38844347000122 Total Reward: 422.6470588235229 Avg_Loss: 7.649013777740863
238
Episode: 147: Time: 13.818714141845703 Total Reward: 385.7692307692258 Avg_Loss: 7.666775330775926
238
Episode: 148: Time: 14.050387144088745 Total Reward: 361.6666666666608 Avg_Loss: 7.414704041320737
238
Episode: 149: Time: 13.788384199142456 Total Reward: 223.02120141343164 Avg_Loss: 7.236904840509431
self.validation_rewards = [383.7412734227116, 462.0183116500987, 447.39722629731267]
Validation Mean Reward: 447.39722629731267 Validation Std Reward: 166.88909154980638
238
Episode: 150: Time: 14.138184070587158 Total Reward: 218.99317406143814 Avg_Loss: 7.393581673377702
238
Episode: 151: Time: 14.288895845413208 Total Reward: 488.3333333333251 Avg_Loss: 7.178153765301745
238
Episode: 152: Time: 14.886891603469849 Total Reward: 129.48979591837136 Avg_Loss: 7.5109160612611205
238
Episode: 153: Time: 13.671997308731079 Total Reward: 498.4065934065852 Avg_Loss: 6.9399019940560605
238
Episode: 154: Time: 13.981750726699829 Total Reward: 329.13793103447927 Avg_Loss: 6.9655071689802055
238
Episode: 155: Time: 13.92575740814209 Total Reward: 326.23287671231935 Avg_Loss: 6.693466865214981
238
Episode: 156: Time: 13.793350219726562 Total Reward: 404.9999999999924 Avg_Loss: 7.173781354888146
238
Episode: 157: Time: 14.309649467468262 Total Reward: 363.46153846153413 Avg_Loss: 6.717696476884249
238
Episode: 158: Time: 13.752549409866333 Total Reward: 536.2056737588568 Avg_Loss: 6.8522568435228175
238
Episode: 159: Time: 13.336105585098267 Total Reward: 298.3823529411736 Avg_Loss: 7.261177442153962
238
Episode: 160: Time: 15.195365190505981 Total Reward: 396.3494809688549 Avg_Loss: 7.216840846197946
238

Episode: 161: Time: 14.078561782836914 Total Reward: 222.56756756757215 Avg_Loss: 6.722673019441236
238
Episode: 162: Time: 13.947440385818481 Total Reward: 367.8378378378334 Avg_Loss: 6.746140658855438
238
Episode: 163: Time: 13.747451066970825 Total Reward: 411.66666666666015 Avg_Loss: 6.845230890923188
238
Episode: 164: Time: 13.902198553085327 Total Reward: 413.6505190311323 Avg_Loss: 7.5823719942269205
238
Episode: 165: Time: 14.730907678604126 Total Reward: 372.69230769230626 Avg_Loss: 6.89247851762451
238
Episode: 166: Time: 14.714712619781494 Total Reward: 341.13707165108906 Avg_Loss: 6.756070050371795
238
Episode: 167: Time: 14.028451442718506 Total Reward: 328.20819112627925 Avg_Loss: 6.778029134544004
238
Episode: 168: Time: 13.939057350158691 Total Reward: 249.48160535116884 Avg_Loss: 6.973816724384532
238
Episode: 169: Time: 13.785015106201172 Total Reward: 466.6438356164306 Avg_Loss: 6.65182393288412
238
Episode: 170: Time: 13.497497081756592 Total Reward: 417.19512195121365 Avg_Loss: 6.811714335649955
238
Episode: 171: Time: 14.617365837097168 Total Reward: 364.94065281898827 Avg_Loss: 6.729505106681535
238
Episode: 172: Time: 14.416533470153809 Total Reward: 465.7142857142802 Avg_Loss: 6.551807728134284
238
Episode: 173: Time: 13.63762092590332 Total Reward: 491.7158671586644 Avg_Loss: 6.782364249229431
238
Episode: 174: Time: 13.334226369857788 Total Reward: 476.4285714285629 Avg_Loss: 6.383254089776208
238
Episode: 175: Time: 13.561361312866211 Total Reward: 388.1460674157271 Avg_Loss: 6.456955951802871
238
Episode: 176: Time: 13.68553900718689 Total Reward: 399.66192170818033 Avg_Loss: 5.906681862698884
238
Episode: 177: Time: 13.547927141189575 Total Reward: 569.2066420664116 Avg_Loss: 6.046334391632
238
Episode: 178: Time: 13.731239557266235 Total Reward: 236.1036789297709 Avg_Loss: 6.47881622124119
238
Episode: 179: Time: 14.14729356765747 Total Reward: 330.74257425742667 Avg_Loss: 6.681830752296608
238
Episode: 180: Time: 14.556043148040771 Total Reward: 321.9278996865171 Avg_Loss: 6.509084878849382
238
Episode: 181: Time: 13.753684520721436 Total Reward: 432.39726027396546 Avg_Loss: 6.53365380072794
238
Episode: 182: Time: 13.93233847618103 Total Reward: 437.64604810995667 Avg_Loss: 6.660794593706853
238
Episode: 183: Time: 14.722710371017456 Total Reward: 476.8390804597635 Avg_Loss: 6.31963626426809
238
Episode: 184: Time: 14.342447280883789 Total Reward: 559.3209876543083 Avg_Loss: 6.449406402952531
238
Episode: 185: Time: 13.711095094680786 Total Reward: 454.81549815497533 Avg_Loss: 6.6643546033306285
238
Episode: 186: Time: 14.099413394927979 Total Reward: 277.61146496815064 Avg_Loss: 6.726235087178335
238
Episode: 187: Time: 14.081170797348022 Total Reward: 368.6363636363587 Avg_Loss: 6.824191791670663
238
Episode: 188: Time: 13.687862873077393 Total Reward: 401.52777777777527 Avg_Loss: 7.06206549566333
238
Episode: 189: Time: 13.601712465286255 Total Reward: 373.30985915492437 Avg_Loss: 6.653621753223804
238
Episode: 190: Time: 13.472751379013062 Total Reward: 637.8519855595564 Avg_Loss: 7.212036154851192
238
Episode: 191: Time: 13.947800636291504 Total Reward: 456.36986301369274 Avg_Loss: 6.696039343080601
238
Episode: 192: Time: 13.981067419052124 Total Reward: 408.1446540880429 Avg_Loss: 7.056630100522723
238
Episode: 193: Time: 13.506483554840088 Total Reward: 638.0827067669045 Avg_Loss: 6.7703606035529065
238
Episode: 194: Time: 14.524585008621216 Total Reward: 341.53250773993705 Avg_Loss: 6.822574484749
238
Episode: 195: Time: 13.714739561080933 Total Reward: 467.0689655172333 Avg_Loss: 6.846578516379124
238
Episode: 196: Time: 15.643826484680176 Total Reward: 394.0510948905074 Avg_Loss: 6.637073029490078
238
Episode: 197: Time: 14.473036527633667 Total Reward: 306.7094017094028 Avg_Loss: 6.89213053168369
238
Episode: 198: Time: 13.730220556259155 Total Reward: 399.38202247190816 Avg_Loss: 6.84395168608978
238
Episode: 199: Time: 14.104511022567749 Total Reward: 440.0318471337469 Avg_Loss: 6.61656884485934
self.validation_rewards = [383.7412734227116, 462.0183116500987, 447.39722629731267, 470.9319037848952]
Validation Mean Reward: 470.9319037848952 Validation Std Reward: 194.75762601780383
Test Mean Reward: 522.9326428821707 Test Std Reward: 173.74290547277437

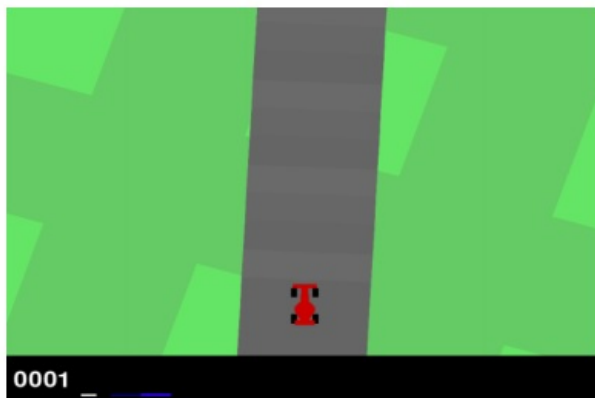
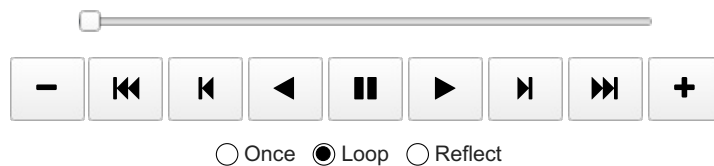
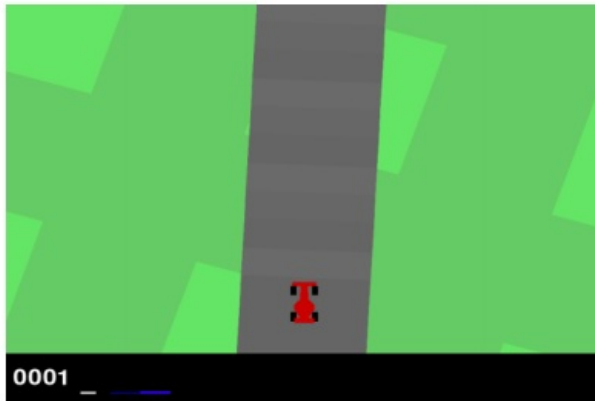


Plots for SoftUpdateDQN ($\tau = 0.01$)



```
In [ ]: total_rewards, frames = trainSoftUpdateDQN.play_episode(0,True,42)
anim = animate(frames)
HTML(anim.to_jshtml())
```

Out[]:



SoftUpdateDQN with $\tau = 0.9$

```
In [ ]: import DQN
import utils
import torch

# tau = 0.9

traineSoftUpdatedQN = DQN.SoftUpdatedQN(EnvWrapper(env),
    model.Nature_Paper_Conv,
    tau = 0.9,
    update_freq = 1,
    lr = 0.00025,
    gamma = 0.95,
    buffer_size=20000,
    batch_size=16,
    loss_fn = "mse_loss",
    use_wandb = False,
    device = 'cpu',
    seed = 42,
    epsilon_scheduler = utils.exponential_decay(1, 1000, 0.1),
    save_path = utils.get_save_path("DoubleDQN_SoftUpdates", "./runs/"))

traineSoftUpdatedQN.train(200, 50, 30, 50, 50)

saving to ./runs/DoubleDQN_SoftUpdates/run2
```

```
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:370: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
```

```
states = torch.tensor(states).clone().detach().to(self.device)
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:371: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
actions = torch.tensor(actions).clone().detach().to(self.device)
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:372: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
rewards = torch.tensor(rewards).clone().detach().to(self.device)
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:373: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
next_states = torch.tensor(next_states).clone().detach().to(self.device)
/content/drive/MyDrive/EE239AS.2/RL-part1/DQN.py:374: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
dones = torch.tensor(dones).clone().detach().float().to(self.device)
```

```
79
Episode: 0: Time: 18.67995309829712 Total Reward: -62.62589928057628 Avg_Loss: 0.7038815219568301
238
Episode: 1: Time: 32.093379497528076 Total Reward: -59.74358974359032 Avg_Loss: 0.5181654855812422
238
Episode: 2: Time: 32.49520540237427 Total Reward: -32.95620437956218 Avg_Loss: 0.6411781224555203
238
Episode: 3: Time: 33.236116886138916 Total Reward: -48.57142857142931 Avg_Loss: 0.5697589866942218
238
Episode: 4: Time: 35.879544734954834 Total Reward: -27.659932659932966 Avg_Loss: 0.6593217390174625
238
Episode: 5: Time: 34.17969036102295 Total Reward: -35.14084507042304 Avg_Loss: 0.6954051946767238
238
Episode: 6: Time: 34.87087607383728 Total Reward: -83.54961832061035 Avg_Loss: 0.6826389205775091
238
Episode: 7: Time: 35.16612792015076 Total Reward: -59.84375000000046 Avg_Loss: 0.6908798629903242
238
Episode: 8: Time: 36.112157344818115 Total Reward: 10.072463768114945 Avg_Loss: 0.7202956523245373
238
Episode: 9: Time: 33.89221906661987 Total Reward: 7.272727272726218 Avg_Loss: 0.8377223849578315
238
Episode: 10: Time: 36.57988739013672 Total Reward: 284.08496732025344 Avg_Loss: 1.0821630739833878
238
Episode: 11: Time: 33.8481068611145 Total Reward: -62.50902527075877 Avg_Loss: 1.577274447416558
238
Episode: 12: Time: 36.77599477767944 Total Reward: 77.20543806646683 Avg_Loss: 1.3778717947982941
238
Episode: 13: Time: 34.25280237197876 Total Reward: 7.389078498294147 Avg_Loss: 1.6150403045305686
238
Episode: 14: Time: 37.569344997406006 Total Reward: 291.4406779660951 Avg_Loss: 1.9240835469936122
238
Episode: 15: Time: 34.48411202430725 Total Reward: -30.691318327974965 Avg_Loss: 1.8732050123835813
238
Episode: 16: Time: 36.12346935272217 Total Reward: 87.45614035087769 Avg_Loss: 2.421976727347414
238
Episode: 17: Time: 39.24706959724426 Total Reward: 291.44067796609295 Avg_Loss: 2.2271193988433406
238
Episode: 18: Time: 35.27979278564453 Total Reward: 126.1838006230567 Avg_Loss: 2.304597315167179
238
Episode: 19: Time: 35.57199573516846 Total Reward: 21.504854368933373 Avg_Loss: 2.534822064412742
238
Episode: 20: Time: 34.43801498413086 Total Reward: -73.87323943661988 Avg_Loss: 2.3959893581747007
238
Episode: 21: Time: 37.576823234558105 Total Reward: 246.2162162162152 Avg_Loss: 2.5412868154900417
238
Episode: 22: Time: 35.629424810409546 Total Reward: 373.08510638296855 Avg_Loss: 3.1517683132355954
238
Episode: 23: Time: 34.20887470245361 Total Reward: 51.95340501792476 Avg_Loss: 2.946369639590007
238
Episode: 24: Time: 35.7877082824707 Total Reward: 190.2564102564143 Avg_Loss: 2.6133272619057104
238
Episode: 25: Time: 34.81828761100769 Total Reward: 505.7326007325939 Avg_Loss: 3.2751836209487513
238
Episode: 26: Time: 35.99305510520935 Total Reward: 118.2867132867176 Avg_Loss: 3.9871022603591952
238
Episode: 27: Time: 35.31793546676636 Total Reward: 380.52447552446876 Avg_Loss: 3.8162284378244093
238
Episode: 28: Time: 38.91838312149048 Total Reward: 278.0407523510897 Avg_Loss: 4.125971066100257
238
Episode: 29: Time: 36.407278060913086 Total Reward: 383.87323943660914 Avg_Loss: 4.173452612482199
238
Episode: 30: Time: 35.90271592140198 Total Reward: 449.80286738349963 Avg_Loss: 4.500980471863466
238
Episode: 31: Time: 37.042322874069214 Total Reward: 5.000000000000529 Avg_Loss: 5.23826371471421
238
Episode: 32: Time: 35.058308839797974 Total Reward: 425.91254752851216 Avg_Loss: 5.065956177581258
238
```

Episode: 33: Time: 38.3540575504303 Total Reward: 166.43790849673604 Avg_Loss: 4.993577953897605
238
Episode: 34: Time: 36.877875328063965 Total Reward: 107.5316455696248 Avg_Loss: 5.570055973880431
238
Episode: 35: Time: 37.1038019657135 Total Reward: 414.4339622641412 Avg_Loss: 4.909677060962725
238
Episode: 36: Time: 36.49241542816162 Total Reward: 177.42524916943947 Avg_Loss: 5.696206484271698
161
Episode: 37: Time: 24.642181634902954 Total Reward: -33.15887850467166 Avg_Loss: 5.4773642761366705
238
Episode: 38: Time: 34.89583086967468 Total Reward: 666.1940298507348 Avg_Loss: 5.246600323865394
238
Episode: 39: Time: 36.393572092056274 Total Reward: 306.89873417720526 Avg_Loss: 5.28893569963319
238
Episode: 40: Time: 36.65230679512024 Total Reward: 221.36363636363478 Avg_Loss: 5.845275898440545
238
Episode: 41: Time: 34.76272487640381 Total Reward: 4.236641221374732 Avg_Loss: 9.981875817064477
238
Episode: 42: Time: 39.309218406677246 Total Reward: 280.0000000000017 Avg_Loss: 8.928152489061116
238
Episode: 43: Time: 36.65817165374756 Total Reward: 512.2727272727157 Avg_Loss: 5.368174809868596
238
Episode: 44: Time: 35.9464430809021 Total Reward: 30.423728813560814 Avg_Loss: 5.6232114113679454
238
Episode: 45: Time: 37.13586497306824 Total Reward: 384.4520547945182 Avg_Loss: 5.182152297316479
238
Episode: 46: Time: 36.11061143875122 Total Reward: 672.8571428571287 Avg_Loss: 8.73708976967996
238
Episode: 47: Time: 35.99907994270325 Total Reward: 179.76038338658617 Avg_Loss: 5.880644922121232
238
Episode: 48: Time: 36.32237458229065 Total Reward: 442.41496598639196 Avg_Loss: 5.887047465608902
238
Episode: 49: Time: 36.94936919212341 Total Reward: 632.8911564625766 Avg_Loss: 5.757988179431242
self.validation_rewards = [17.203634465456496]
Validation Mean Reward: 17.203634465456496 Validation Std Reward: 276.17138991857996
238
Episode: 50: Time: 36.92998385429382 Total Reward: 320.54054054054143 Avg_Loss: 6.626788958531468
238
Episode: 51: Time: 36.720309019088745 Total Reward: 400.4128440366914 Avg_Loss: 6.447348163408392
238
Episode: 52: Time: 35.18409609794617 Total Reward: 551.2585034013503 Avg_Loss: 5.337813329796831
238
Episode: 53: Time: 37.73522400856018 Total Reward: 396.03942652329016 Avg_Loss: 6.716917973356087
238
Episode: 54: Time: 34.77240824699402 Total Reward: 735.8823529411644 Avg_Loss: 9.918661759180182
238
Episode: 55: Time: 36.243030071258545 Total Reward: 570.467625899267 Avg_Loss: 6.312868244507733
238
Episode: 56: Time: 34.572542667388916 Total Reward: 730.9109311740747 Avg_Loss: 9.183043814506851
238
Episode: 57: Time: 35.95697355270386 Total Reward: 488.038869257946 Avg_Loss: 6.850746144266689
238
Episode: 58: Time: 35.504806995391846 Total Reward: 298.5018050541433 Avg_Loss: 6.246813611323092
238
Episode: 59: Time: 34.714402198791504 Total Reward: 561.2499999999944 Avg_Loss: 6.505636089990119
238
Episode: 60: Time: 37.78161931037903 Total Reward: 351.88644688644035 Avg_Loss: 7.100777233849053
238
Episode: 61: Time: 35.71400499343872 Total Reward: 363.0645161290312 Avg_Loss: 7.0042127976898385
238
Episode: 62: Time: 36.89695072174072 Total Reward: 498.333333333245 Avg_Loss: 8.708591117578393
238
Episode: 63: Time: 36.45803666114807 Total Reward: 431.50176678444365 Avg_Loss: 8.7854191054817
238
Episode: 64: Time: 35.143293619155884 Total Reward: -31.241610738255744 Avg_Loss: 7.170437602936721
238
Episode: 65: Time: 37.80713367462158 Total Reward: 525.2531645569527 Avg_Loss: 7.035952912909644
238
Episode: 66: Time: 36.350908041000366 Total Reward: 315.25641025640874 Avg_Loss: 6.762904542834819
238
Episode: 67: Time: 35.651880741119385 Total Reward: 521.949152542366 Avg_Loss: 7.267999525831527
238
Episode: 68: Time: 37.65113306045532 Total Reward: 68.70106761566069 Avg_Loss: 6.986939351849196
238
Episode: 69: Time: 34.379520654678345 Total Reward: 317.2448979591861 Avg_Loss: 6.631112786902099
238
Episode: 70: Time: 36.28809380531311 Total Reward: 578.0038022813602 Avg_Loss: 9.023017848239226
238
Episode: 71: Time: 34.74528455734253 Total Reward: 379.63768115941593 Avg_Loss: 7.490538771913833
238
Episode: 72: Time: 36.53744411468506 Total Reward: 318.7931034482663 Avg_Loss: 6.893106600817512
238
Episode: 73: Time: 36.43398880958557 Total Reward: 390.53054662378884 Avg_Loss: 7.464504379184306
238
Episode: 74: Time: 35.25416421890259 Total Reward: 540.4166666666575 Avg_Loss: 8.019195823609328
238
Episode: 75: Time: 38.26566958427429 Total Reward: 57.24913494810021 Avg_Loss: 7.934464906944948
238
Episode: 76: Time: 35.34301471710205 Total Reward: 180.00000000000392 Avg_Loss: 9.235168373384395

238
Episode: 77: Time: 36.3647096157074 Total Reward: 357.45901639343424 Avg_Loss: 7.632101574364831
238
Episode: 78: Time: 37.30892300605774 Total Reward: 236.03448275862442 Avg_Loss: 7.61900752532382
238
Episode: 79: Time: 35.009284019470215 Total Reward: 461.66666666666567 Avg_Loss: 7.041030262197767
238
Episode: 80: Time: 36.820356607437134 Total Reward: 243.46153846154172 Avg_Loss: 8.489074419025613
238
Episode: 81: Time: 35.84219455718994 Total Reward: 273.5897435897434 Avg_Loss: 7.291245907795529
238
Episode: 82: Time: 35.34237456321716 Total Reward: 281.7605633802749 Avg_Loss: 7.425390150366711
238
Episode: 83: Time: 37.12748193740845 Total Reward: 238.33333333332504 Avg_Loss: 8.482884875866546
238
Episode: 84: Time: 35.564202547073364 Total Reward: 329.3323442136501 Avg_Loss: 10.002250637326922
238
Episode: 85: Time: 36.163982629776 Total Reward: 327.79411764705327 Avg_Loss: 7.926747521432508
238
Episode: 86: Time: 35.31170964241028 Total Reward: 302.39413680782013 Avg_Loss: 8.857719036210485
238
Episode: 87: Time: 36.125168323516846 Total Reward: 115.52631578947648 Avg_Loss: 7.960466974422712
238
Episode: 88: Time: 36.28199553489685 Total Reward: 203.7012987012999 Avg_Loss: 8.138407744279428
238
Episode: 89: Time: 35.37859845161438 Total Reward: 344.8496240601405 Avg_Loss: 8.732806125608812
238
Episode: 90: Time: 38.409905672073364 Total Reward: 224.74921630094246 Avg_Loss: 9.077914396253954
238
Episode: 91: Time: 34.83888053894043 Total Reward: 470.8914728682147 Avg_Loss: 7.310220507513575
238
Episode: 92: Time: 36.63164782524109 Total Reward: 509.3956043955984 Avg_Loss: 7.667894937911956
238
Episode: 93: Time: 36.29645800590515 Total Reward: 223.49315068493556 Avg_Loss: 7.545929531089398
238
Episode: 94: Time: 36.29892826080322 Total Reward: 174.73684210526534 Avg_Loss: 7.264702591575494
238
Episode: 95: Time: 37.64824891090393 Total Reward: 376.63120567375364 Avg_Loss: 7.093824316974447
238
Episode: 96: Time: 38.219594955444336 Total Reward: 408.08641975308433 Avg_Loss: 8.192391997625847
238
Episode: 97: Time: 37.615504026412964 Total Reward: 207.8169014084549 Avg_Loss: 7.829437363548439
238
Episode: 98: Time: 38.10647368431091 Total Reward: 249.4108761329331 Avg_Loss: 7.947372424752772
238
Episode: 99: Time: 37.01971435546875 Total Reward: 164.6491228070215 Avg_Loss: 7.902699570195014
self.validation_rewards = [17.203634465456496, 348.4530306528225]
Validation Mean Reward: 348.4530306528225 Validation Std Reward: 87.52186673486187
238
Episode: 100: Time: 37.40770196914673 Total Reward: 437.2033898304985 Avg_Loss: 7.453907722184638
238
Episode: 101: Time: 36.19826602935791 Total Reward: 354.47735191637526 Avg_Loss: 8.296416821600008
238
Episode: 102: Time: 36.404579401016235 Total Reward: 519.3344709897501 Avg_Loss: 8.518967973584889
238
Episode: 103: Time: 37.50993776321411 Total Reward: 354.3243243243231 Avg_Loss: 7.714001755754487
238
Episode: 104: Time: 36.169506549835205 Total Reward: 414.93377483442714 Avg_Loss: 7.318849949275746
238
Episode: 105: Time: 36.46966481208801 Total Reward: 410.3003533568878 Avg_Loss: 7.588292154444366
238
Episode: 106: Time: 37.77203583717346 Total Reward: 333.1690140845059 Avg_Loss: 7.814265818155112
238
Episode: 107: Time: 37.910974740982056 Total Reward: 364.55882352941086 Avg_Loss: 7.584156533750165
238
Episode: 108: Time: 36.11357641220093 Total Reward: 408.3783783783762 Avg_Loss: 8.640705011471981
238
Episode: 109: Time: 37.30412220954895 Total Reward: 404.9999999999944 Avg_Loss: 8.100879311561584
238
Episode: 110: Time: 36.749802589416504 Total Reward: 467.68221574343704 Avg_Loss: 8.20272875833912
238
Episode: 111: Time: 33.281824827194214 Total Reward: -39.80519480519551 Avg_Loss: 8.194937149015795
238
Episode: 112: Time: 35.10482740402222 Total Reward: 173.9655172413825 Avg_Loss: 8.885378886671628
238
Episode: 113: Time: 36.1713593006134 Total Reward: 424.99999999999466 Avg_Loss: 8.389181493210192
238
Episode: 114: Time: 39.86853218078613 Total Reward: 218.84615384615745 Avg_Loss: 8.293746396273123
238
Episode: 115: Time: 37.99055790901184 Total Reward: 302.3941368078149 Avg_Loss: 8.165428890901453
238
Episode: 116: Time: 36.04036211967468 Total Reward: 94.09090909091269 Avg_Loss: 7.375404035844722
238
Episode: 117: Time: 38.068645000457764 Total Reward: 273.07817589576797 Avg_Loss: 7.5503073179421305
238
Episode: 118: Time: 37.6317412853241 Total Reward: 415.2739726027341 Avg_Loss: 7.019992664581587
238
Episode: 119: Time: 35.94101357460022 Total Reward: 97.85714285714306 Avg_Loss: 7.559968582722319
238

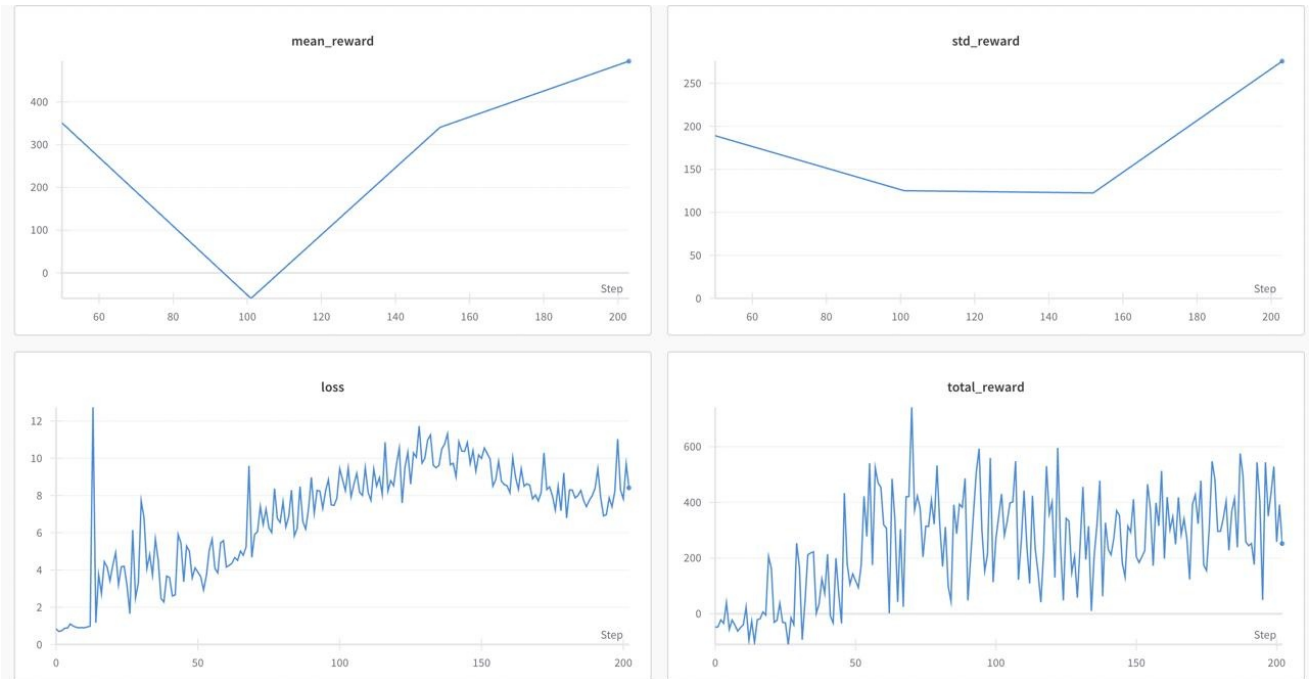
Episode: 120: Time: 37.65149188041687 Total Reward: 493.815789473676 Avg_Loss: 6.985345752299333
238
Episode: 121: Time: 39.832311391830444 Total Reward: 366.76470588235094 Avg_Loss: 7.501509400976806
238
Episode: 122: Time: 37.85736322402954 Total Reward: 395.44585987260956 Avg_Loss: 8.266160539218358
238
Episode: 123: Time: 39.612807512283325 Total Reward: 362.9124579124516 Avg_Loss: 8.150460627900452
238
Episode: 124: Time: 39.83660316467285 Total Reward: 455.31446540879284 Avg_Loss: 7.416995963128675
238
Episode: 125: Time: 39.17702341079712 Total Reward: 261.90235690235505 Avg_Loss: 7.875604189744516
238
Episode: 126: Time: 40.356898069381714 Total Reward: 382.0642201834817 Avg_Loss: 7.89894899750958
238
Episode: 127: Time: 39.53917360305786 Total Reward: 258.50318471336846 Avg_Loss: 7.427404134714303
238
Episode: 128: Time: 39.01865863800049 Total Reward: 485.85808580857326 Avg_Loss: 8.154242996408158
238
Episode: 129: Time: 41.338868141174316 Total Reward: 189.21052631579317 Avg_Loss: 7.492306886099968
238
Episode: 130: Time: 38.43080139160156 Total Reward: 293.692579505297 Avg_Loss: 8.390125433436962
238
Episode: 131: Time: 40.19451713562012 Total Reward: 311.55737704917857 Avg_Loss: 7.118398120924204
238
Episode: 132: Time: 38.36785554885864 Total Reward: 248.94904458599163 Avg_Loss: 7.874670574144155
238
Episode: 133: Time: 42.01980710029602 Total Reward: 385.5653710247288 Avg_Loss: 7.622773362808869
238
Episode: 134: Time: 39.37663769721985 Total Reward: 306.3840830449791 Avg_Loss: 6.681554274899619
238
Episode: 135: Time: 39.757951736450195 Total Reward: 350.69288389513196 Avg_Loss: 7.939154864359303
238
Episode: 136: Time: 35.82477021217346 Total Reward: 621.5354330708577 Avg_Loss: 6.700654980515232
238
Episode: 137: Time: 36.92944860458374 Total Reward: 470.7370517928175 Avg_Loss: 8.287526612021342
238
Episode: 138: Time: 37.93934774398804 Total Reward: 289.14634146341785 Avg_Loss: 7.825808929795978
238
Episode: 139: Time: 37.359344482421875 Total Reward: 67.57668711656804 Avg_Loss: 7.135783943809381
238
Episode: 140: Time: 37.4439001083374 Total Reward: 157.7777777777789 Avg_Loss: 6.91705904638066
238
Episode: 141: Time: 36.77516150474548 Total Reward: 442.93103448274917 Avg_Loss: 7.009214863055894
238
Episode: 142: Time: 38.256407499313354 Total Reward: 414.9999999999919 Avg_Loss: 7.557238206142137
238
Episode: 143: Time: 39.373841762542725 Total Reward: 176.64179104477967 Avg_Loss: 7.969437768980234
238
Episode: 144: Time: 37.618093729019165 Total Reward: 454.15254237287644 Avg_Loss: 7.097863185305555
238
Episode: 145: Time: 36.49379920959473 Total Reward: 261.55737704918295 Avg_Loss: 7.002640779779739
238
Episode: 146: Time: 37.16029667854309 Total Reward: 213.8235294117662 Avg_Loss: 6.81500635577851
238
Episode: 147: Time: 37.66323757171631 Total Reward: 395.38461538461297 Avg_Loss: 6.514232002386526
238
Episode: 148: Time: 37.725725412368774 Total Reward: 548.3333333333244 Avg_Loss: 6.432330228200479
238
Episode: 149: Time: 37.590503215789795 Total Reward: 442.1024734982241 Avg_Loss: 6.921407837827666
self.validation_rewards = [17.203634465456496, 348.4530306528225, 546.1872942217469]
Validation Mean Reward: 546.1872942217469 Validation Std Reward: 229.8915859343739
238
Episode: 150: Time: 38.268775939941406 Total Reward: 492.03071672353724 Avg_Loss: 7.29087466452302
238
Episode: 151: Time: 37.46394944190979 Total Reward: 529.9999999999928 Avg_Loss: 6.944536131971023
238
Episode: 152: Time: 39.05230164527893 Total Reward: 234.44606413994597 Avg_Loss: 6.886536087308611
238
Episode: 153: Time: 37.826584339141846 Total Reward: 340.8974358974332 Avg_Loss: 7.1152465573879855
238
Episode: 154: Time: 38.176717042922974 Total Reward: 146.37931034483188 Avg_Loss: 7.18705299271255
238
Episode: 155: Time: 37.93672251701355 Total Reward: 288.56164383561025 Avg_Loss: 7.393285014048344
238
Episode: 156: Time: 36.93415307998657 Total Reward: 546.7910447761083 Avg_Loss: 6.369275110609391
238
Episode: 157: Time: 38.06878447532654 Total Reward: 397.3076923076883 Avg_Loss: 7.1820240907308435
238
Episode: 158: Time: 37.782029151916504 Total Reward: 490.1063829787198 Avg_Loss: 7.7300909216664415
238
Episode: 159: Time: 38.97583556175232 Total Reward: 324.117647058818 Avg_Loss: 7.313919887823217
238
Episode: 160: Time: 37.75984477996826 Total Reward: 282.16262975778295 Avg_Loss: 7.031136521271297
125
Episode: 161: Time: 18.886272192001343 Total Reward: 2.3270270270288904 Avg_Loss: 7.166693706512451
238
Episode: 162: Time: 38.08798313140869 Total Reward: 350.94594594594656 Avg_Loss: 7.312863351918068
238
Episode: 163: Time: 38.01358985900879 Total Reward: 308.3333333333354 Avg_Loss: 7.168958123491592

238
 Episode: 164: Time: 38.14614939689636 Total Reward: 424.0311418685101 Avg_Loss: 10.185948565727523
 238
 Episode: 165: Time: 38.37383961677551 Total Reward: 283.4615384615407 Avg_Loss: 7.6234201207882215
 238
 Episode: 166: Time: 37.67885398864746 Total Reward: 378.520249221182 Avg_Loss: 7.58094951785913
 238
 Episode: 167: Time: 40.029919147491455 Total Reward: 386.2286689419726 Avg_Loss: 9.548939423400816
 238
 Episode: 168: Time: 37.82853412628174 Total Reward: 15.367892976588067 Avg_Loss: 7.5311501597156045
 238
 Episode: 169: Time: 34.90076422691345 Total Reward: 620.7534246575217 Avg_Loss: 7.947609636963916
 238
 Episode: 170: Time: 34.40687298774719 Total Reward: 514.7560975609656 Avg_Loss: 8.969762463529571
 238
 Episode: 171: Time: 36.49416446685791 Total Reward: 347.1364985163182 Avg_Loss: 8.259223440615068
 238
 Episode: 172: Time: 34.28822422027588 Total Reward: 383.5714285714232 Avg_Loss: 8.242902969111915
 238
 Episode: 173: Time: 35.20144867897034 Total Reward: 384.7047970479649 Avg_Loss: 7.882830429477852
 238
 Episode: 174: Time: 36.14343976974487 Total Reward: 512.142857142851 Avg_Loss: 7.155737664018359
 238
 Episode: 175: Time: 33.869564056396484 Total Reward: 283.2771535580431 Avg_Loss: 7.574166123105698
 238
 Episode: 176: Time: 35.372788429260254 Total Reward: 481.5124555160077 Avg_Loss: 8.810277470019685
 238
 Episode: 177: Time: 34.39322304725647 Total Reward: 550.7564575645674 Avg_Loss: 7.304138942425992
 238
 Episode: 178: Time: 36.09143614768982 Total Reward: 292.95986622073895 Avg_Loss: 8.216597672270126
 238
 Episode: 179: Time: 35.96487069129944 Total Reward: 479.2574257425679 Avg_Loss: 7.616665769024055
 238
 Episode: 180: Time: 35.47803092002869 Total Reward: 375.2194357366718 Avg_Loss: 7.3812606262058775
 238
 Episode: 181: Time: 36.26068902015686 Total Reward: 504.31506849314235 Avg_Loss: 6.470830105933823
 238
 Episode: 182: Time: 37.22158408164978 Total Reward: 410.15463917525034 Avg_Loss: 7.955303294318063
 238
 Episode: 183: Time: 35.4965283870697 Total Reward: 209.59770114942887 Avg_Loss: 7.566282724632936
 238
 Episode: 184: Time: 35.63231372833252 Total Reward: 674.5473251028698 Avg_Loss: 8.370038021512393
 238
 Episode: 185: Time: 35.920313119888306 Total Reward: 510.1660516605081 Avg_Loss: 7.3149705774643845
 238
 Episode: 186: Time: 35.57785701751709 Total Reward: 287.16560509553364 Avg_Loss: 7.1085509242129925
 238
 Episode: 187: Time: 36.93478274345398 Total Reward: 444.3939393939295 Avg_Loss: 7.493375180148277
 238
 Episode: 188: Time: 36.189356327056885 Total Reward: 616.8055555555434 Avg_Loss: 7.68924972690454
 238
 Episode: 189: Time: 36.19288754463196 Total Reward: 556.4084507042162 Avg_Loss: 7.019930305100289
 238
 Episode: 190: Time: 36.50150012969971 Total Reward: 13.303249097473897 Avg_Loss: 7.7374506527636235
 238
 Episode: 191: Time: 36.55545401573181 Total Reward: 381.02739726026624 Avg_Loss: 7.198674059715591
 238
 Episode: 192: Time: 35.39353966712952 Total Reward: 417.57861635219354 Avg_Loss: 7.236925419639139
 238
 Episode: 193: Time: 35.59972262382507 Total Reward: 374.9248120300713 Avg_Loss: 7.215984680071599
 238
 Episode: 194: Time: 36.8681161403656 Total Reward: 338.43653250773923 Avg_Loss: 7.214087282409187
 238
 Episode: 195: Time: 33.83772325515747 Total Reward: 494.6551724137832 Avg_Loss: 7.538623120103564
 238
 Episode: 196: Time: 34.84542226791382 Total Reward: 488.9416058394072 Avg_Loss: 7.394926351158559
 238
 Episode: 197: Time: 36.31237983703613 Total Reward: 241.18233618233882 Avg_Loss: 8.017909611974444
 238
 Episode: 198: Time: 34.30575346946716 Total Reward: 489.2696629213376 Avg_Loss: 7.339842819867014
 238
 Episode: 199: Time: 35.508596420288086 Total Reward: 178.8853503184758 Avg_Loss: 7.433680044502771

```
In [ ]: import DQN
import utils
import torch

trainSoftUpdatedQDN = DQN.SoftUpdateDQN(EnvWrapper(env),
    model.Nature_Paper_Conv,
    tau = 0.9,
    update_freq = 1,
    lr = 0.00025,
    gamma = 0.95,
    buffer_size=20000,
    batch_size=16,
    loss_fn = "mse_loss",
    use_wandb = False,
    device = 'cpu',
    seed = 42,
```

```
epsilon_scheduler = utils.exponential_decay(1, 1000, 0.1),
save_path = "/content/drive/MyDrive/EE239AS.2/RL-part1/runs/DoubleDQN_SoftUpdates/run2")
```



```
In [ ]: traineSoftUpdateDQN.load_model(suffix = 'best')
traineSoftUpdateDQN.validate(10)
```

```
Out[ ]: (558.3495506114696, 134.59302297315028)
```

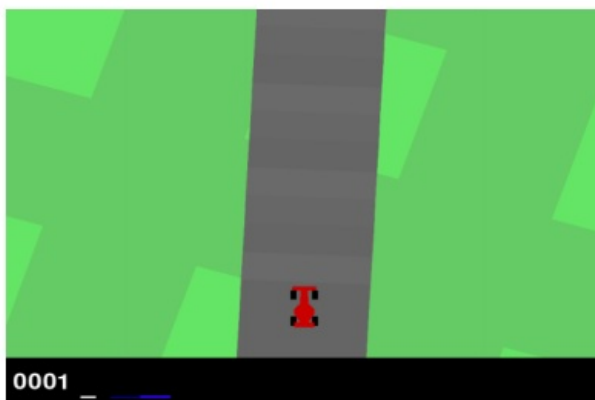
The test mean reward for Soft Update DQN with updated τ

= 0.9 is 558.34

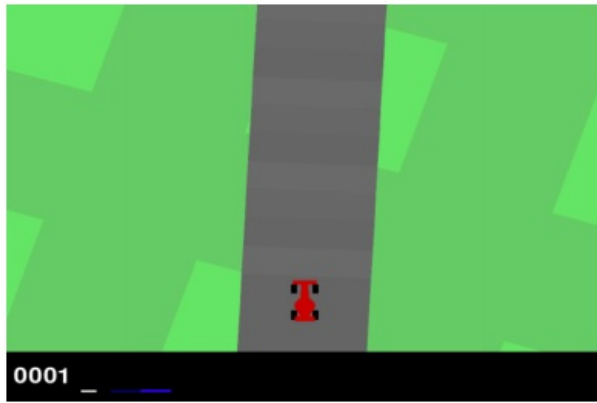
558.34

```
In [ ]: total_rewards, frames = traineSoftUpdateDQN.play_episode(0, True, 42)
anim = animate(frames)
HTML(anim.to_jshtml())
```

```
Out[ ]:
```



☐ Once ☒ Loop ☐ Reflect



Questions:

- Which method performed better? (5 points)
- If we modify the τ

for soft updates or the C

for the hard updates, how does this affect the performance of the model, come up with a intuition for this, then experimentally verify this. (5 points)

Answers

| Model | Parameter | Test Mean Reward |
|---------------|---------------|--------------------------|
| Vanilla DQN | - | 596.4705857600395 |
| HardUpdateDQN | $C=100$ | 505.7495048573211 |
| | $C=1$ | 561.0064446524763 |
| SoftUpdateDQN | $\tau = 0.01$ | 522.9326428821707 |
| | $\tau = 0.9$ | 558.3495506114696 |

- From above table, when in comparison between only the Double DQNs, we note that HardUpdateDQN (~ 561) performs very slightly better than Hard Update (~ 558). Overall amongst all the three, the Vanilla DQN seems to perform better, because all the models are trained for the same amount of episodes. Our understanding is that if SoftUpdateDQN is allowed to train for more episodes, owing to it being stable in nature, it could outperform the other two models.

Varying τ in SoftUpdateDQN

- We try two τ

values for our analysis with So i.e τ

$= 0.01$ and τ

$= 0.9 : \backslash$

Intution :

1. Small τ (0.01): This means the target network is updated very slowly. This provides very stable targets, which can be crucial in environments with noisy or high variance rewards. However, it might slow down the learning process because the target network does not adapt quickly to the changes in the current network.
2. Large τ (0.9): This means the target network is updated almost as quickly as the current network. This can lead to faster learning as the target network can adapt quickly to the current network's improvements. However, it can also lead to instability in training if the updates are too aggressive, causing the targets to shift rapidly.

Varying τ in SoftUpdateDQN

Experimental Results :

We obtain the following test mean rewards:

1. $\tau = 0.01$ gave a test mean reward of nearly 522.

2. $\tau = 0.9$ gave a test mean reward of nearly 558.

Interpretation :

1. $\tau = 0.01$ (Small τ):

- The target network is updated very slowly.
- This leads to very stable targets, reducing the variance in the Q-value targets.
- As a result, the training process is stable but slower because the target network does not quickly adapt to the latest Q-value improvements.
- The test mean reward of 522 indicates that the agent learned a reasonably good policy but might not have fully exploited the latest improvements in the Q-network.

2. $\tau = 0.9$ (Large τ):

- The target network is updated quickly.
- This leads to less stable targets but allows the target network to adapt quickly to the changes in the Q-network.
- This can accelerate the learning process since the agent gets feedback from the target network that reflects the latest Q-value improvements.
- The higher test mean reward of 558 suggests that the agent was able to learn a better policy faster, possibly because the targets were more aligned with the current Q-network.

The intuition and experimental results suggest that:

- Smaller τ values lead to more stable but slower learning.
- Larger τ values lead to faster learning but can introduce instability.

In our DQN experiments, $\tau = 0.9$ provided better performance, indicating that the environment and model can handle faster updates without significant instability. However, this might not generalize to all environments.

Varying C (update_freq) in HardUpdateDQN

In a HardUpdate Deep Q-Network (DQN), the hyperparameter `update_freq` (often denoted as (C)) controls how often the weights of the target network are updated to match the weights of the online network. Understanding how this parameter affects the learning process and the resulting mean rewards can be elucidated by exploring the roles of the target and online networks in DQN.

Intuition Behind `update_freq`

Stability vs. Flexibility

- **Low `update_freq` (e.g., `update_freq = 1`)**: The target network is updated very frequently, making it nearly identical to the online network most of the time. This can lead to a very responsive but less stable learning process, as the target values used for training are changing frequently.
- **High `update_freq` (e.g., `update_freq = 100`)**: The target network is updated less frequently, providing a more stable set of target values for a longer period. This can improve stability in learning because the target values don't shift as rapidly. However, it can also slow down the learning process because the target network might lag behind the online network significantly, especially in fast-changing environments.

Impact on Mean Rewards

- With a lower `update_freq`, the agent may be able to adjust more quickly to changes in the environment, potentially leading to higher immediate rewards but with a risk of instability and divergent behavior.
- With a higher `update_freq`, the agent benefits from more stable training targets, which can lead to more consistent but potentially slower improvements in policy, possibly resulting in lower immediate rewards compared to a lower `update_freq`.

Analysis of Experimental Results

1. **Mean Reward with `update_freq = 1` :**

- **Test Mean Reward = 561.0064446524763**
- Here, the frequent updates of the target network allow the agent to adapt quickly to new information. This can lead to high short-term performance as observed. However, it also risks the possibility of less stable learning and potentially larger variances in performance.

2. **Mean Reward with `update_freq = 100` :**

- **Test Mean Reward = 505.7495048573211**
- With less frequent updates, the target network provides more stable training targets, which can contribute to a more stable learning process. The slightly lower mean reward indicates that while the learning is more stable, the agent may not adapt as

rapidly to new information compared to the case with `update_freq = 1`.

Matching Experimental Results with Intuition

The results you observed align well with the intuitive expectations:

- **Higher Mean Reward with `update_freq = 1`**: The more frequent updates allow the agent to quickly incorporate new information, leading to higher immediate performance.
- **Lower Mean Reward with `update_freq = 100`**: The less frequent updates contribute to a more stable learning process, but the slower adaptation results in a lower immediate performance.

Conclusion

The choice of `update_freq` involves a trade-off between stability and adaptability. A low `update_freq` can lead to higher mean rewards in the short term due to quick adaptability, but at the cost of potential instability. A higher `update_freq` can result in a more stable learning process with potentially lower immediate rewards due to slower adaptation. The optimal `update_freq` depends on the specific environment and the balance between stability and adaptability that yields the best long-term performance.

env_wrapper.py

```
In [ ]: import cv2
import numpy as np
import gymnasium as gym
import matplotlib.pyplot as plt
from utils import preprocess #this is a helper function that may be useful to grayscale and crop the image

class EnvWrapper(gym.Wrapper):
    def __init__(
        self,
        env:gym.Env,
        skip_frames:int=4,
        stack_frames:int=4,
        initial_no_op:int=50,
        do_nothing_action:int=0,
        **kwargs
    ):
        """the environment wrapper for CarRacing-v2

        Args:
            env (gym.Env): the original environment
            skip_frames (int, optional): the number of frames to skip, in other words we will
            repeat the same action for `skip_frames` steps. Defaults to 4.
            stack_frames (int, optional): the number of frames to stack, we stack
            `stack_frames` frames to form the state and allow agent understand the motion of the car. Defaults
            initial_no_op (int, optional): the initial number of no-op steps to do nothing at the beginning of
            do_nothing_action (int, optional): the action index for doing nothing. Defaults to 0, which should
            discretization of the action space.
        """
        super(EnvWrapper, self).__init__(env, **kwargs)
        self.initial_no_op = initial_no_op
        self.skip_frames = skip_frames
        self.stack_frames = stack_frames
        self.observation_space = gym.spaces.Box(
            low=0,
            high=1,
            shape=(stack_frames, 84, 84),
            dtype=np.float32
        )
        self.do_nothing_action = do_nothing_action

    def reset(self, **kwargs):
        s, info = self.env.reset(**kwargs)
        for i in range(self.initial_no_op):
            s, r, terminated, truncated, info = self.env.step(self.do_nothing_action)
        s = preprocess(s)
        self.stacked_state = np.tile(s, (self.stack_frames, 1, 1))
        return self.stacked_state, info

    def step(self, action):
        reward = 0
        for _ in range(self.skip_frames):
            s, r, terminated, truncated, info = self.env.step(action)
            reward += r
            if terminated or truncated:
                break
        s = preprocess(s)
```

```

self.stacked_state = np.concatenate((self.stacked_state[1:],
s[np.newaxis]), axis=0)
return self.stacked_state, reward, terminated, truncated, info

```

model.py

```

In [ ]: import torch as torch
import torch.nn as nn

import torch
import torch.nn as nn
import numpy as np

class MLP(nn.Module):
    def __init__(self, input_size:int, action_size:int, hidden_size:int=512,non_linear:nn.Module=nn.ReLU):
        """
        input: tuple[int]
            The input size of the image, of shape (channels, height, width)
        action_size: int
            The number of possible actions
        hidden_size: int
            The number of neurons in the hidden layer

        This is a separate class because it may be useful for the bonus questions
        """
        super(MLP, self).__init__()
        #===== TODO: =====
        self.linear1 = nn.Linear(input_size, hidden_size)
        self.output = nn.Linear(hidden_size, action_size)
        self.non_linear = non_linear()

    def forward(self, x:torch.Tensor)->torch.Tensor:
        #===== TODO: =====
        x = self.linear1(x)
        x = self.non_linear(x)
        x = self.output(x)

        return x

class Nature_Paper_Conv(nn.Module):
    """
    A class that defines a neural network with the following architecture:
    - 1 convolutional layer with 32 8x8 kernels with a stride of 4x4 w/ ReLU activation
    - 1 convolutional layer with 64 4x4 kernels with a stride of 2x2 w/ ReLU activation
    - 1 convolutional layer with 64 3x3 kernels with a stride of 1x1 w/ ReLU activation
    - 1 fully connected layer with 512 neurons and ReLU activation.
    Based on 2015 paper 'Human-level control through deep reinforcement learning' by Mnih et al
    """
    def __init__(self, input_size:tuple[int], action_size:int,**kwargs):
        """
        input: tuple[int]
            The input size of the image, of shape (channels, height, width)
        action_size: int
            The number of possible actions
        **kwargs: dict
            additional kwargs to pass for stuff like dropout, etc if you would want to implement it
        """
        super(Nature_Paper_Conv, self).__init__()
        #===== TODO: =====
        self.CNN = nn.Sequential(
            nn.Conv2d(input_size[0], 32, kernel_size=8, stride=4),
            nn.ReLU(),
            nn.Conv2d(32, 64, kernel_size=4, stride=2),
            nn.ReLU(),
            nn.Conv2d(64, 64, kernel_size=3, stride=1),
            nn.ReLU()
        )
        self.MLP = MLP(self._conv_output_size(input_size), action_size)

    def forward(self, x:torch.Tensor)->torch.Tensor:
        #===== TODO: =====
        x = self.CNN(x)
        x = x.view(x.size(0), -1) # Flatten the output from convolutions
        x = self.MLP(x)
        return x

    def _conv_output_size(self, shape):
        """
        Helper function to calculate the output size of the convolutional layers.
        """
        dummy_input = torch.zeros(1, *shape)
        dummy_output = self.CNN(dummy_input)
        return dummy_output.view(-1).size(0)

```


DQN.py

```
In [ ]: import torch
import torch.optim as optim
import torch.nn.functional as F
import torch.nn
import gymnasium as gym
from replay_buffer import ReplayBufferDQN
import wandb
import random
import numpy as np
import os
import time
from utils import exponential_decay
import typing
from matplotlib import pyplot as plt

class DQN:
    def __init__(self, env: typing.Union[gym.Env, gym.Wrapper],
                 #model params
                 model: torch.nn.Module,
                 model_kwargs: dict = {},
                 #overall hyperparams
                 lr: float = 0.001, gamma: float = 0.99,
                 buffer_size: int = 10000, batch_size: int = 32,
                 loss_fn: str = 'mse_loss',
                 use_wandb: bool = False, device: str = 'cpu',
                 seed: int = 42,
                 epsilon_scheduler = exponential_decay(1, 700, 0.1),
                 save_path: str = None):
        """Initializes the DQN algorithm

        Args:
            env (gym.Env|gym.Wrapper): the environment to train on
            model (torch.nn.Module): the model to train
            model_kwargs (dict, optional): the keyword arguments to pass to the model. Defaults to {}.
            lr (float, optional): the learning rate to use in the optimizer. Defaults to 0.001.
            gamma (float, optional): discount factor. Defaults to 0.99.
            buffer_size (int, optional): the size of the replay buffer. Defaults to 10000.
            batch_size (int, optional): the batch size. Defaults to 32.
            loss_fn (str, optional): the name of the loss function to use. Defaults to 'mse_loss'.
            use_wandb (bool, optional): _description_. Defaults to False.
            device (str, optional): _description_. Defaults to 'cpu'.
            seed (int, optional): the seed to use for reproducibility. Defaults to 42.
            epsilon_scheduler ([type], optional): the epsilon scheduler to use, must have a __call__ method tha
            save_path (str, optional): _description_. Defaults to None.

        Raises:
            ValueError: _description_
        """

        self.env = env
        self._set_seed(seed)

        self.observation_space = self.env.observation_space.shape
        self.model = model(
            self.observation_space,
            self.env.action_space.n, **model_kwargs
        ).to(device)
        self.model.train()
        self.optimizer = optim.Adam(self.model.parameters(), lr = lr)
        self.gamma = gamma

        self.replay_buffer = ReplayBufferDQN(buffer_size)
        self.batch_size = batch_size
        self.i_update = 0
        self.device = device
        self.epsilon_decay = epsilon_scheduler
        self.save_path = save_path if save_path is not None else ""

        #set the loss function
        if loss_fn == 'smooth_l1_loss':
            self.loss_fn = F.smooth_l1_loss
        elif loss_fn == 'mse_loss':
            self.loss_fn = F.mse_loss
        else:
            raise ValueError('loss_fn must be either smooth_l1_loss or mse_loss')

        self.wandb = use_wandb
        if self.wandb:
            wandb.init(project = 'racing-car-dqn')
            #log the hyperparameters
            wandb.config.update({
                'lr': lr,
                'gamma': gamma,
                'buffer_size': buffer_size,
```

```

        'batch_size': batch_size,
        'loss_fn': loss_fn,
        'device': device,
        'seed': seed,
        'save_path': save_path
    })

#####
# Add lists to store metrics
self.episode_rewards = []
self.episode_losses = []
self.validation_rewards = []
self.validation_stds = []
#####

def train(self, n_episodes:int = 1000, validate_every:int = 100, n_validation_episodes:int = 10, n_test_episodes:int = 10):
    os.makedirs(self.save_path, exist_ok = True)
    best_val_reward = -np.inf

    for episode in range(n_episodes):
        state, _ = self.env.reset()
        done = False
        truncated = False
        total_reward = 0
        i = 0
        loss = 0
        start_time = time.time()
        epsilon = self.epsilon_decay()
        while (not done) and (not truncated):
            action = self.sample_action(state, epsilon)
            next_state, reward, done, truncated, _ = self.env.step(action)
            self.replay_buffer.add(state, action, reward, next_state, done)
            total_reward += reward
            state = next_state

            not_warm_starting, l = self._optimize_model()
            if not_warm_starting:
                loss += l
                epsilon = self.epsilon_decay()
                i += 1

        print(i)
        if i > 0:
            avg_loss = loss / i
        else:
            avg_loss = 0

        #####
        # Log metrics to lists
        self.episode_rewards.append(total_reward)
        self.episode_losses.append(avg_loss)
        #####

        if self.wandb:
            wandb.log({'total_reward': total_reward, 'loss': avg_loss})

        print(f"Episode: {episode}: Time: {time.time() - start_time} Total Reward: {total_reward} Avg Loss: {avg_loss}")
        if episode % validate_every == validate_every - 1:
            mean_reward, std_reward = self.validate(n_validation_episodes)
            self.validation_rewards.append(mean_reward)
            self.validation_stds.append(std_reward)
            print("self.validation_rewards = ", self.validation_rewards)
            if self.wandb:
                wandb.log({'mean_reward': mean_reward, 'std_reward': std_reward})
            print("Validation Mean Reward: {} Validation Std Reward: {}".format(mean_reward, std_reward))
            if mean_reward > best_val_reward:
                best_val_reward = mean_reward
                self._save('best')

        if episode % save_every == save_every - 1:
            self._save(str(episode))

    self._save('final')
    self.load_model('best')
    mean_reward, std_reward = self.validate(n_test_episodes)
    if self.wandb:
        wandb.log({'mean_test_reward': mean_reward, 'std_test_reward': std_reward})
    print("Test Mean Reward: {} Test Std Reward: {}".format(mean_reward, std_reward))

    # Plot the metrics
    self.plot_metrics()

#####
def plot_metrics(self):
    # Plot total rewards
    plt.figure(figsize=(12, 5))
    plt.subplot(2, 2, 1)
    plt.plot(self.episode_rewards, label='Total Reward')

```

```

plt.xlabel('Episode')
plt.ylabel('Total Reward')
plt.title('Total Reward per Episode')
plt.legend()

# Plot average losses
plt.subplot(2, 2, 2)
plt.plot(self.episode_losses, label='Average Loss', color='orange')
plt.xlabel('Episode')
plt.ylabel('Average Loss')
plt.title('Average Loss per Episode')
plt.legend()

# Plot validation rewards
plt.subplot(2, 2, 3)
plt.plot(self.validation_rewards, label="Val Mean Rewards")
plt.xlabel("Validation Step")
plt.ylabel("Mean Reward")
plt.legend()

# Plot validation std deviations
plt.subplot(2, 2, 4)
plt.plot(self.validation_stds, label="Val Std Dev")
plt.xlabel("Validation Step")
plt.ylabel("Standard Deviation")
plt.legend()

plt.tight_layout()
plt.show()

#####
def _optimize_model(self):
    """Optimizes the model

    Returns:
        bool: whether we have enough samples to optimize the model, which we define as having at least 10*batch_size
        float: the loss, if we do not have enough samples, we return 0
    """
    #===== TODO: =====
    if len(self.replay_buffer) < 10 * self.batch_size:
        return False, 0

    states, actions, rewards, next_states, dones = self.replay_buffer.sample(self.batch_size)

    states = torch.tensor(states).clone().detach().to(self.device)
    actions = torch.tensor(actions).clone().detach().to(self.device)
    rewards = torch.tensor(rewards).clone().detach().to(self.device)
    next_states = torch.tensor(next_states).clone().detach().to(self.device)
    dones = torch.tensor(dones).clone().detach().float().to(self.device)

    q_values = self.model(states).gather(1, actions.unsqueeze(1)).squeeze(1)
    next_q_values = self.model(next_states).max(1)[0]
    target_q_values = rewards + (1 - dones) * self.gamma * next_q_values

    loss = self.loss_fn(q_values, target_q_values.detach())

    self.optimizer.zero_grad()
    loss.backward()
    self.optimizer.step()
    return True, loss.item()

def _sample_action(self, state:np.ndarray
                    , epsilon:float = 0.1)->int:
    """Samples an action from the model

    Args:
        state (np.ndarray): the state, of shape [n_c,h,w]
        epsilon (float, optional): the epsilon for epsilon greedy. Defaults to 0.1.

    Returns:
        int: the index of the action to take
    """
    #===== TODO: =====
    if random.random() < epsilon:
        return self.env.action_space.sample()
    else:
        state = torch.tensor(state, dtype=torch.float32).unsqueeze(0).to(self.device)
        with torch.no_grad():
            q_values = self.model(state)
        return q_values.argmax().item()

def _set_seed(self, seed:int):

```

```

random.seed(seed)
np.random.seed(seed)
self.seed = seed
torch.manual_seed(seed)
torch.cuda.manual_seed(seed)
torch.backends.cudnn.deterministic = True
gym.utils.seeding.np_random(seed)

def _validate_once(self):
    state, _ = self.env.reset()
    # print(state)
    done = False
    truncated = False
    total_reward = 0
    i = 0
    # epsilon = self.epsilon_decay()
    while (not done) and (not truncated):
        action = self._sample_action(state, 0)
        # out = self.env.step(action)
        next_state, reward, done, truncated, _ = self.env.step(action)
        # next_state = np.array(state_buffer[-self.n_frames:])
        total_reward += reward
        state = next_state
    return total_reward

def validate(self, n_episodes:int = 10):
    # self.model.eval()
    rewards_per_episode = []
    for _ in range(n_episodes):
        rewards_per_episode.append(self._validate_once())
    # self.model.train()
    return np.mean(rewards_per_episode), np.std(rewards_per_episode)

def load_model(self, suffix:str = ''):
    self.model.load_state_dict(torch.load(os.path.join(self.save_path, f'model_{suffix}.pt')))

def _save(self, suffix:str = ''):
    torch.save(self.model.state_dict(), os.path.join(self.save_path, f'model_{suffix}.pt'))

def play_episode(self, epsilon:float = 0, return_frames:bool = True, seed:int = None):
    """Plays an episode of the environment

    Args:
        epsilon (float, optional): the epsilon for epsilon greedy. Defaults to 0.
        return_frames (bool, optional): whether we should return frames. Defaults to True.
        seed (int, optional): the seed for the enviroment. Defaults to None.

    Returns:
        if return_frames is True, returns the total reward and the frames
        if return_frames is False, returns the total reward
    """
    if seed is not None:
        state, _ = self.env.reset(seed = seed)
    else:
        state, _ = self.env.reset()

    done = False
    total_reward = 0
    if return_frames:
        frames = []

    with torch.no_grad():
        while not done:
            action = self._sample_action(state, epsilon)
            next_state, reward, terminated, truncated, _ = self.env.step(action)
            total_reward += reward
            done = terminated or truncated
            if return_frames:
                frames.append(self.env.render())
            state = next_state

    if return_frames:
        return total_reward, frames

    return total_reward

```

class HardUpdatedDQN(DQN):

```

def __init__(self, env, model, model_kwargs:dict = {},
             update_freq:int = 5, *args, **kwargs):
    super().__init__(env, model, model_kwargs, *args, **kwargs)
    #===== TODO: =====
    self.update_freq = update_freq
    self.target_model = model(
        self.observation_space,
        self.env.action_space.n, **model_kwargs
    )

```

```

).to(self.device)
self.target_model.load_state_dict(self.model.state_dict())
self.target_model.eval()

def _optimize_model(self):
    """Optimizes the model

    Returns:
        bool: whether we have enough samples to optimize the model, which we define as having at least 10*batch_size
        float: the loss, if we do not have enough samples, we return 0
    """
    #===== TODO: =====
    if len(self.replay_buffer) < 10 * self.batch_size:
        return False, 0

    states, actions, rewards, next_states, dones = self.replay_buffer.sample(self.batch_size)

    states = torch.tensor(states).clone().detach().to(self.device)
    actions = torch.tensor(actions).clone().detach().to(self.device)
    rewards = torch.tensor(rewards).clone().detach().to(self.device)
    next_states = torch.tensor(next_states).clone().detach().to(self.device)
    dones = torch.tensor(dones).clone().detach().float().to(self.device)

    Q_expected = self.model(states).gather(1, actions.unsqueeze(1)).squeeze(1)

    Q_targets_next = self.target_model(next_states).detach().max(1)[0]

    Q_targets = rewards + self.gamma * Q_targets_next * (1 - dones)
    # print(Q_expected.shape, Q_targets.shape)

    loss = self.loss_fn(Q_expected, Q_targets)
    self.optimizer.zero_grad()
    loss.backward()
    self.optimizer.step()
    self._update_model()

    return True, loss.item()

    #hint: you can copy over most of the code from the parent class
    #and only change one line

def _update_model(self):
    self.i_update += 1
    if self.i_update % self.update_freq == 0:
        self.target_model.load_state_dict(self.model.state_dict())

def _save(self, suffix:str = ''):
    torch.save(self.model.state_dict(), os.path.join(self.save_path, f'model_{suffix}.pt'))
    torch.save(self.target_model.state_dict(), os.path.join(self.save_path, f'target_model_{suffix}.pt'))

def load_model(self, suffix:str = ''):
    self.model.load_state_dict(torch.load(os.path.join(self.save_path, f'model_{suffix}.pt')))
    self.target_model.load_state_dict(torch.load(os.path.join(self.save_path, f'target_model_{suffix}.pt')))

class SoftUpdatedQN(HardUpdatedQN):
    def __init__(self, env, model, model_kwargs=dict = {},
                 tau:float = 0.01, *args, **kwargs):
        super().__init__(env, model, model_kwargs, *args, **kwargs)
        self.tau = tau

    def _update_model(self):
        """Soft updates the target model"""
        #===== TODO: =====
        for target_param, param in zip(self.target_model.parameters(), self.model.parameters()):
            target_param.data.copy_(self.tau * param.data + (1.0 - self.tau) * target_param.data)

```

```

In [ ]: #@title Convert ipynb to HTML in Colab
# Upload ipynb
from google.colab import files
f = files.upload()

# Convert ipynb to html
import subprocess
file0 = list(f.keys())[0]
_ = subprocess.run(["pip", "install", "nbconvert"])
_ = subprocess.run(["jupyter", "nbconvert", file0, "--to", "html"])

# download the html
files.download(file0[:-5]+"html")

```

Choose Files No file selected

Upload widget is only available when the cell has been executed in the

current browser session. Please rerun this cell to enable.
Saving Project4_RL.ipynb to Project4_RL.ipynb


```
In [ ]: %load_ext autoreload
        %autoreload 2
        %env MUJOCO_GL=egl
```

```
In [ ]: !pip install -q numpy torch wandb swig gymnasium[mujoco] matplotlib termcolor
```

```
In [ ]: from google.colab import drive
        drive.mount("/content/drive/")

        # add system path to current directory
        import sys
        sys.path.insert(1, '/content/drive/MyDrive/EE239AS.2/RL-part2')
        %cd "/content/drive/MyDrive/EE239AS.2/RL-part2"
```

```
In [ ]: import test
        from utils import *
```

Reinforcement Learning Part 2: DDPG

By Lawrence Liu

Some General Instructions

- This entire assignment will be worth 5 points of extra credit for project 4, and will be due on the same day as project 4, so June 7th.
- You will be implementing a DDPG agent to solve the DoublePendulum environment.
- Because this is a bonus, there will be no test cases.
- You will need to implement the TODOs in the `ddpg.py` and `model.py` files. DO NOT use Windows for this project, gymnasium does is not supported for windows and installing it will be difficult.

Introduction to the Enviroment

We will be training a DDPG agent to solve the DoublePendulum environment. The DoublePendulum environment is a classic control problem where the goal is to balance a double pendulum on a cart.

Action Space

The agent can apply a force to the cart in the range of -1 to 1. This is a continuous action space.

Observation Space

The observation space is a 11 dimensional vector. The first 1 is the position of the cart, the next 4 are the cosines and sins of different angles of the double pendulum, and the next 3 are the velocities of the cart and the pendulum, and the final 3 are the constrain forces on the pendulum. You can find more information about these constraint forces [here](#)

Reward

The reward can be decomposed into 3 parts. The first part is an alive bonus that pays +10 for every time step the second pendulum is upright. There are 2 penalty terms, one for the tip of the second pendulum moving too much, and another for the cart moving too fast.

You can find more information about the environment [here](#)

```
In [ ]: import gymnasium as gym

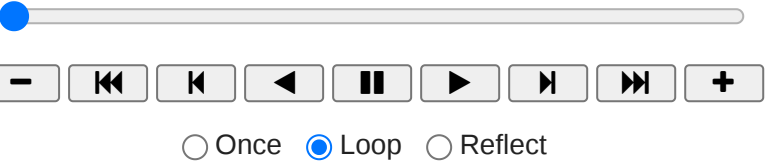
        import numpy as np
        env = gym.make("InvertedDoublePendulum-v4")
        env.seed(np.random.RandomState(42))
```

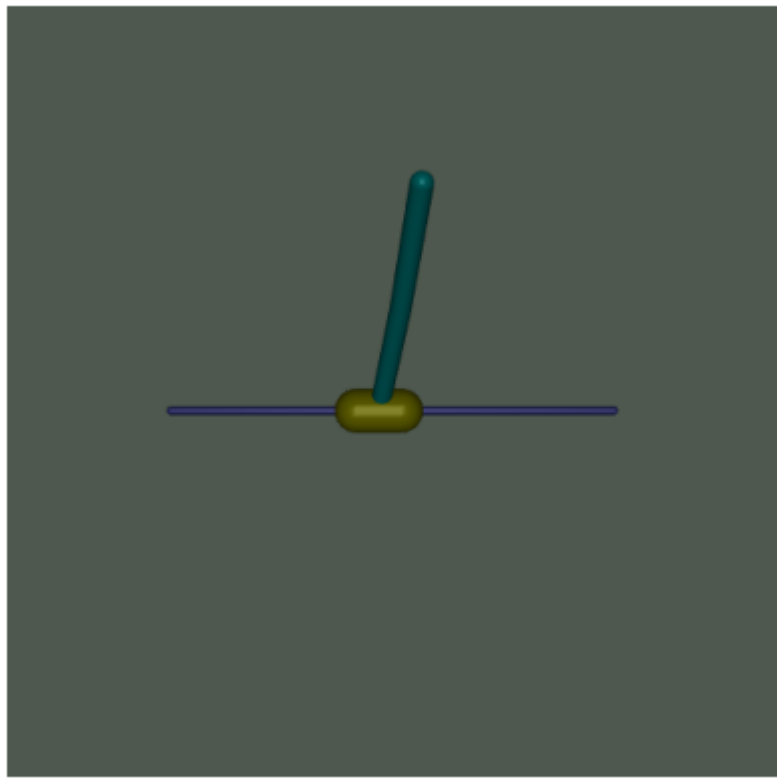
```
from IPython.display import HTML

frames = []
s, _ = eval_env.reset()

while True:
    a = eval_env.action_space.sample()
    s, r, terminated, truncated, _ = eval_env.step(a)
    frames.append(eval_env.render())
    if terminated or truncated:
        break

anim = animate(frames)
HTML(anim.to_jshtml())
```





Model (1 point)

Because the inputs to the model is a 11 dimensional vector, we will use a MLP. Specifically we will follow the architecture in the DDPG paper. For DDPG we have both an Actor and a Critic. The Actor is responsible for selecting the action, and the Critic is responsible for evaluating the action.

Actor

The Actor is a 3 layer MLP:

- Layer 1: 400 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of $-1/\sqrt{\text{fan_in}}$ to $1/\sqrt{\text{fan_in}}$
- Layer 2: 300 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of $-1/\sqrt{\text{fan_in}}$ to $1/\sqrt{\text{fan_in}}$
- Layer 3: 1 unit, tanh activation, intialized with uniform weights in the range of -0.003 to 0.003 ##### Critic The Critic is a 3 layer MLP:
- Layer 1: 400 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of $-1/\sqrt{\text{fan_in}}$ to $1/\sqrt{\text{fan_in}}$
- Layer 2: 300 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of $-1/\sqrt{\text{fan_in}}$ to $1/\sqrt{\text{fan_in}}$. Input is the concatenation of the 400 dimension embedding from the state, and the action taken.
- Layer 3: 1 unit, intialized with uniform weights in the range of -0.003 to 0.003

```
In [ ]: import model

In [ ]: import torch as torch
import torch.nn as nn
import numpy as np
import torch.nn.functional as F

def fanin_init(size, fanin=None):
    #a helper function to initialize the weights of the model
    fanin = fanin or size[0]
    v = 1. / np.sqrt(fanin)
    return torch.Tensor(size).uniform_(-v, v)

class Actor(nn.Module):
    """Actor model for the DDPG algorithm.

    Layer 1: 400 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of -1/sqrt(fan_in) to 1/sqrt(fan_in)
    Layer 2: 300 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of -1/sqrt(fan_in) to 1/sqrt(fan_in)
    Layer 3: 1 unit, tanh activation, intialized with uniform weights in the range of -0.003 to 0.003
```

```

"""
def __init__(self, input_size:tuple[int], action_size:int,CNN = None):
    """
    input: tuple[int]
        The input size
    action_size: int
        The number of actions
    """
    super(Actor, self).__init__()
    self.fc1 = nn.Linear(np.prod(input_size), 400)
    self.non_linear = nn.ReLU()

    self.fc2 = nn.Linear(400, 300)
    self.fc3 = nn.Linear(300, action_size)
    self.tanh = nn.Tanh()

def init_weights(self,init_w=3e-3):
    """

    Args:
        init_w (float, optional): the onesided range of the uniform distribution for the final layer. Defaults to 3e-3.

    """
    #initialize the weights of the model
    self.fc1.weight.data = fanin_init(self.fc1.weight.data.size())
    self.fc2.weight.data = fanin_init(self.fc2.weight.data.size())
    self.fc3.weight.data.uniform_(-init_w, init_w)

def forward(self, x:torch.Tensor)->torch.Tensor:
    x = self.non_linear(self.fc1(x))
    x = self.non_linear(self.fc2(x))
    return self.tanh(self.fc3(x))

class Critic(nn.Module):
    """Critic model for the DDPG algorithm.
    Layer 1: 400 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of -1/sqrt(fan_in) to 1/sqrt(fan_in)
    Layer 2: 300 units, ReLU activation, Fan-in weight initialization, ie each weight is initialized with a uniform distribution in the range of -1/sqrt(fan_in) to 1/sqrt(fan_in). Input i
    Layer 3: 1 unit, intialized with uniform weights in the range of -0.003 to 0.003
    """
    def __init__(self,input_size:tuple[int],action_size:int):
        """
        input: tuple[int]
            The input size of the state
        action_size: int
            The number of actions
        """
        super(Critic, self).__init__()
        self.fc1 = nn.Linear(np.prod(input_size), 400)
        self.fc2 = nn.Linear(400+action_size, 300)
        self.fc3 = nn.Linear(300, 1)
        self.non_linear = nn.ReLU()

def init_weights(self,init_w=3e-3):
    #initialize the weights of the model
    self.fc1.weight.data = fanin_init(self.fc1.weight.data.size())
    self.fc2.weight.data = fanin_init(self.fc2.weight.data.size())
    self.fc3.weight.data.uniform_(-init_w, init_w)

def forward(self, x:torch.Tensor, a:torch.Tensor)->torch.Tensor:
    x = self.non_linear(self.fc1(x))
    x = self.non_linear(self.fc2(torch.cat([x,a],1)))
    return self.fc3(x)

```

Exploration (1 point)

Because DDPG is an off policy algorithm, we will use a noise process to encourage exploration. Specifically we will use the Ornstein-Uhlenbeck process. The Ornstein-Uhlenbeck process is a stochastic process that generates temporally correlated noise. The process is defined by the following stochastic differential equation:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t$$

Where θ is the rate of mean reversion, μ is the long run mean of the process, σ is the volatility of the process, and W_t is a Wiener process. We can discretize this process to get the following:

$$x_{t+1} = x_t + \theta(\mu - x_t)dt + \sigma\sqrt{dt}\mathcal{N}(0,1)$$

Where $\mathcal{N}(0,1)$ is a sample from the standard normal distribution. We will asume that our steps are of unit length, so we can simplify this to:

$$x_{t+1} = x_t + \theta(\mu - x_t) + \sigma\mathcal{N}(0,1)$$

We will use $\theta = 0.15$, $\mu = 0$, and $\sigma = 0.2$. We will add this to our action in the following way

$$a_t = \min(\max(\mu(s_t) + x_t, -1), 1)$$

Where a_t is the action taken by the agent, $\mu(s_t)$ is the action selected by the actor, and x_t is the noise generated by the Ornstein-Uhlenbeck process. Please implement the `OU_Noise` class in DDPG.py

DDPG.py

```
In [ ]: import torch
import torch.optim as optim
import torch.nn.functional as F
import torch.nn
import gymnasium as gym
from replay_buffer import ReplayBufferDDPG
import wandb
import random
import numpy as np
import os
import time
import model
from utils import *
import tqdm

class OU_Noise:
    def __init__(self, action_space:int, action_range:list[np.ndarray[float]],
                 mu:float = 0.0, theta:float = 0.15, sigma:float = 0.2, seed:int = 42):
        """Initialize the OU noise

        Args:
            action_space (int): The size of the action space
            action_range (list[np.ndarray[float]]): The range of the action space, the first
                element is the lower bound and the second element is the upper bound
            mu (float, optional): average of the noise. Defaults to 0.0.
            theta (float, optional): the speed of mean reversion. Defaults to 0.15.
            sigma (float, optional): the volatility of the noise. Defaults to 0.2.
            seed (int, optional): the seed for the random number generator. Defaults to 42.
        """
        self.action_space = action_space
        self.mu = mu
        self.theta = theta
        self.sigma = sigma
        self.x = np.ones(self.action_space) * self.mu
        self.action_range = action_range
        self.seed = seed
        np.random.seed(self.seed)

    def reset(self, sigma:float = 0.2):
```

```

"""Reset the noise

Args:
    sigma (float, optional): you can change the sigma of the noise. Defaults to 0.2.
"""
self.x = np.ones(self.action_space) * self.mu
self.sigma = sigma

def _sample(self):
    """sample the noise per the discretized Ornstein-Uhlenbeck process detailed in the notebook"""
    delta_x = self.theta * (self.mu - self.x) + self.sigma * np.random.randn(self.action_space)
    self.x += delta_x
    return self.x

def noise(self, action: np.ndarray[float]):
    """Add the noise to the action

    Args:
        action (np.ndarray[float]): the action to add the noise to

    Returns:
        noised_action (np.ndarray[float]): the noised action, clipped to the action range
    """
    #you can use the _sample method to get the noise
    action_noised = action + self._sample()
    #clip the action to the action range
    action_noised = np.clip(action_noised, self.action_range[0], self.action_range[1])
    return action_noised

class DDPG:
    def __init__(self, env: gym.Env,
                 actor_model: model.Actor,
                 critic_model: model.Critic,
                 actor_kwargs = {},
                 critic_kwargs = {},
                 actor_lr: float = 0.0001,
                 critic_lr: float = 0.001,
                 gamma: float = 0.99,
                 tau: float = 0.001,
                 buffer_size: int = 10**6,
                 batch_size: int = 64,
                 loss_fn: str = 'mse_loss',
                 use_wandb: bool = False, device: str = 'cpu',
                 seed: int = 42,
                 save_path: str = None):
        """Initialize the DDPG agent

    Args:
        env (_type_): The environment to train on
        actor_model (_type_): the class for the actor model
        critic_model (_type_): the class for the critic model
        actor_kwargs (dict, optional): Additional actor_kwargs . Defaults to {}.
        critic_kwargs (dict, optional): Additional critic_kwargs. Defaults to {}.
        actor_lr (float, optional): The learning rate for the optimizer. Defaults to 0.0001.
        critic_lr (float, optional): The learning rate for the critic optimizer. Defaults to 0.001.
        gamma (float, optional): discount factor. Defaults to 0.99.
        tau (float, optional): soft update parameter for the target networks. Defaults to 0.001.
        buffer_size (int, optional): the size of the replay buffer. Defaults to 10^6
        batch_size (int, optional): the batch size for training. Defaults to 64.
        loss_fn (str, optional): name of the loss function to use. Defaults to 'mse_loss'.
        use_wandb (bool, optional): whether to use wandb. Defaults to False.
        device (str, optional): which device to use. Defaults to 'cpu'.
        seed (int, optional): seed for reproducibility. Defaults to 42.
        save_path (str, optional): path to save the model. Defaults to None.

    """
    self.env = env

```

```

self._set_seed(seed)
self.observation_space = self.env.observation_space.shape
self.actor = actor_model(self.observation_space, self.env.action_space.shape[0]
                          , **actor_kwargs).to(device)
self.critic = critic_model(self.observation_space, self.env.action_space.shape[0]
                           , **critic_kwargs).to(device)

self.target_actor = actor_model(self.observation_space, self.env.action_space.shape[0]
                                , **actor_kwargs).to(device)
self.target_critic = critic_model(self.observation_space, self.env.action_space.shape[0]
                                  , **critic_kwargs).to(device)

self.OU_noise = OU_Noise(self.env.action_space.shape[0],
                         [self.env.action_space.low, self.env.action_space.high])

#sync the target networks with the main networks
self.target_actor.load_state_dict(self.actor.state_dict())
self.target_critic.load_state_dict(self.critic.state_dict())

self.actor_optimizer = optim.Adam(self.actor.parameters(), lr = actor_lr)
self.critic_optimizer = optim.Adam(self.critic.parameters(), lr = critic_lr)
self.gamma = gamma

self.replay_buffer = ReplayBufferDDPG(buffer_size)
self.batch_size = batch_size
self.tau = tau
self.device = device
self.save_path = save_path if save_path is not None else "./"

#set the loss function
if loss_fn == 'smooth_l1_loss':
    self.loss_fn = F.smooth_l1_loss
elif loss_fn == 'mse_loss':
    self.loss_fn = F.mse_loss
else:
    raise ValueError('loss_fn must be either smooth_l1_loss or mse_loss')

self.wandb = use_wandb
if self.wandb:
    wandb.init(project = 'double_pendulum_ddpg')
    #log the hyperparameters
    wandb.config.update({
        'actor_lr': actor_lr,
        'critic_lr': critic_lr,
        'gamma': gamma,
        'buffer_size': buffer_size,
        'batch_size': batch_size,
        'loss_fn': loss_fn,
        'tau': tau,
        'device': device,
        'seed': seed,
        'save_path': save_path
    })

def _set_seed(self, seed:int):
    random.seed(seed)
    np.random.seed(seed)
    self.seed = seed
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True
    gym.utils.seeding.np_random(seed)

def play_episode(self, sigma:float = 0, return_frames:bool = False, seed:int = None, env = None):
    """Play an episode of the environment

    Args:
        sigma (float, optional): the sigma for the OU noise. Defaults to 0.

```

```

        return_frames (bool, optional): whether to return the frames. Defaults to False.
        seed (int, optional): the seed for the environment. Defaults to None.
    """
    if env is None:
        env = self.env
    if seed is not None:
        state, _ = env.reset(seed = seed)
    else:
        state, _ = env.reset()

    if sigma > 0:
        self.OU_noise.reset(sigma)
    done = False
    total_reward = 0
    if return_frames:
        frames = []
    with torch.no_grad():
        while not done:
            action = self.actor(torch.tensor(state).float().to(self.device).unsqueeze(0)).cpu().numpy()[0]
            if sigma > 0:
                action = self.OU_noise.noise(action)
            next_state, reward, terminated, truncated, _ = env.step(action)
            total_reward += reward
            done = terminated or truncated
            if return_frames:
                frames.append(env.render())
            state = next_state
    if return_frames:
        return total_reward, frames
    else:
        return total_reward

def _train_one_batch(self, batch_size):
    """train the agent on a single batch"""
    #TODO:
    # sample the batch
    states, actions, rewards, next_states, dones = self.replay_buffer.sample(batch_size)

    # compute the target Q value
    with torch.no_grad():
        next_actions = self.target_actor(next_states)
        next_Q_values = self.target_critic(next_states, next_actions)

    target_Q_values = rewards.unsqueeze(1) + (self.gamma * next_Q_values * (1 - dones.unsqueeze(1).float()))

    # compute the current Q value
    current_Q_values = self.critic(states, actions)

    # compute the critic loss
    critic_loss = self.loss_fn(current_Q_values, target_Q_values.detach())

    # optimize the critic
    self.critic_optimizer.zero_grad()
    critic_loss.backward()
    self.critic_optimizer.step()

    # compute the actor loss
    actions_pred = self.actor(states)
    actor_loss = -self.critic(states, actions_pred).mean()

    # optimize the actor
    self.actor_optimizer.zero_grad()
    actor_loss.backward()
    self.actor_optimizer.step()

    # update the model

```

```

self._update_model()

# return the losses
return critic_loss.item(), actor_loss.item()

def _update_model(self):
    #TODO:
    for target_param, param in zip(self.target_critic.parameters(), self.critic.parameters()):
        target_param.data.copy_(self.tau * param.data + (1.0 - self.tau) * target_param.data)
    for target_param, param in zip(self.target_actor.parameters(), self.actor.parameters()):
        target_param.data.copy_(self.tau * param.data + (1.0 - self.tau) * target_param.data)

def train(self, episodes:int, val_freq:int, val_episodes:int, test_episodes:int, save_every:int,
          train_every:int = 1):
    """Train the agent

    Args:
        episodes (int): the number of episodes to train for
        val_freq (int): the frequency of validation
        val_episodes (int): the number of episodes to validate for
        test_episodes (int): the number of episodes to test for
        save_every (int): the frequency of saving the model
        train_every (int, optional): the frequency of training per enviroment interaction. Defaults to 1.
    """
    best_val_mean = -np.inf
    for i in range(episodes):
        start_time = time.time()
        # print(sigma)
        state, _ = self.env.reset()
        self.OU_noise.reset()
        done = False
        total_reward = 0
        Q_loss_total = 0
        actor_loss_total = 0
        l = 0

        while not done:
            #TODO:

            with torch.no_grad():
                action = self.actor(torch.tensor(state).float().to(self.device).unsqueeze(0)).cpu().detach().numpy()[0]

            # add the noise
            action = self.OU_noise.noise(action)

            # get the transition
            next_state, reward, terminated, truncated, _ = self.env.step(action)
            done = terminated or truncated

            # store the transition
            self.replay_buffer.add(state, action, reward, next_state, done)

            # update the state
            state = next_state
            total_reward += reward

            l += 1

            # if the replay buffer is large enough, and it is time to train the model
            if len(self.replay_buffer) > self.batch_size and l % train_every == 0:
                Q_loss, actor_loss = self._train_one_batch(self.batch_size)
                Q_loss_total += Q_loss
                actor_loss_total += actor_loss

        if self.wandb:
            wandb.log({
                'total_reward': total_reward,
                'Q_loss': Q_loss_total,
                'actor_loss': actor_loss_total
            })

```

```

    })
    print(f"Episode {i}: Time: {time.time()-start_time}, Total Reward: {total_reward}, Q Loss: {Q_loss_total}, Actor Loss: {actor_loss_total}")

    if i % val_freq == val_freq-1:
        val_mean, val_std = self.validate(val_episodes)

        if self.wandb:
            wandb.log({
                'val_mean': val_mean,
                'val_std': val_std
            })
            print(f"Validation Mean: {val_mean}, Validation Std: {val_std}")
            if val_mean > best_val_mean:
                best_val_mean = val_mean
                self.save_model('best')
            # print("save_every", save_every, i, save_every-1)
            if i % save_every == save_every-1:
                self.save_model(i)
                print("saving model")

self.save_model('final')
self.load_model('best')

test_mean, test_std = self.validate(test_episodes)
print(f"Test Mean: {test_mean}, Test Std: {test_std}")
if self.wandb:
    wandb.log({
        'test_mean': test_mean,
        'test_std': test_std
    })

def validate(self, episodes:int):
    rewards = []
    for _ in range(episodes):
        rewards.append(self.play_episode())
    mean_reward = np.mean(rewards)
    std_reward = np.std(rewards)
    return mean_reward, std_reward

def save_model(self, name:str):
    actor_path = os.path.join(self.save_path, f"actor_{name}.pt")
    actor_model_path = os.path.join(self.save_path, f"actor_target_{name}.pt")
    torch.save(self.actor.state_dict(), actor_path)
    torch.save(self.target_actor.state_dict(), actor_model_path)


    critic_path = os.path.join(self.save_path, f"critic_{name}.pt")
    critic_model_path = os.path.join(self.save_path, f"critic_target_{name}.pt")
    torch.save(self.critic.state_dict(), critic_path)
    torch.save(self.target_critic.state_dict(), critic_model_path)

def load_model(self, name:str):
    actor_path = os.path.join(self.save_path, f"actor_{name}.pt")
    actor_model_path = os.path.join(self.save_path, f"actor_target_{name}.pt")
    self.actor.load_state_dict(torch.load(actor_path))
    self.target_actor.load_state_dict(torch.load(actor_model_path))

    critic_path = os.path.join(self.save_path, f"critic_{name}.pt")
    critic_model_path = os.path.join(self.save_path, f"critic_target_{name}.pt")
    self.critic.load_state_dict(torch.load(critic_path))
    self.target_critic.load_state_dict(torch.load(critic_model_path))

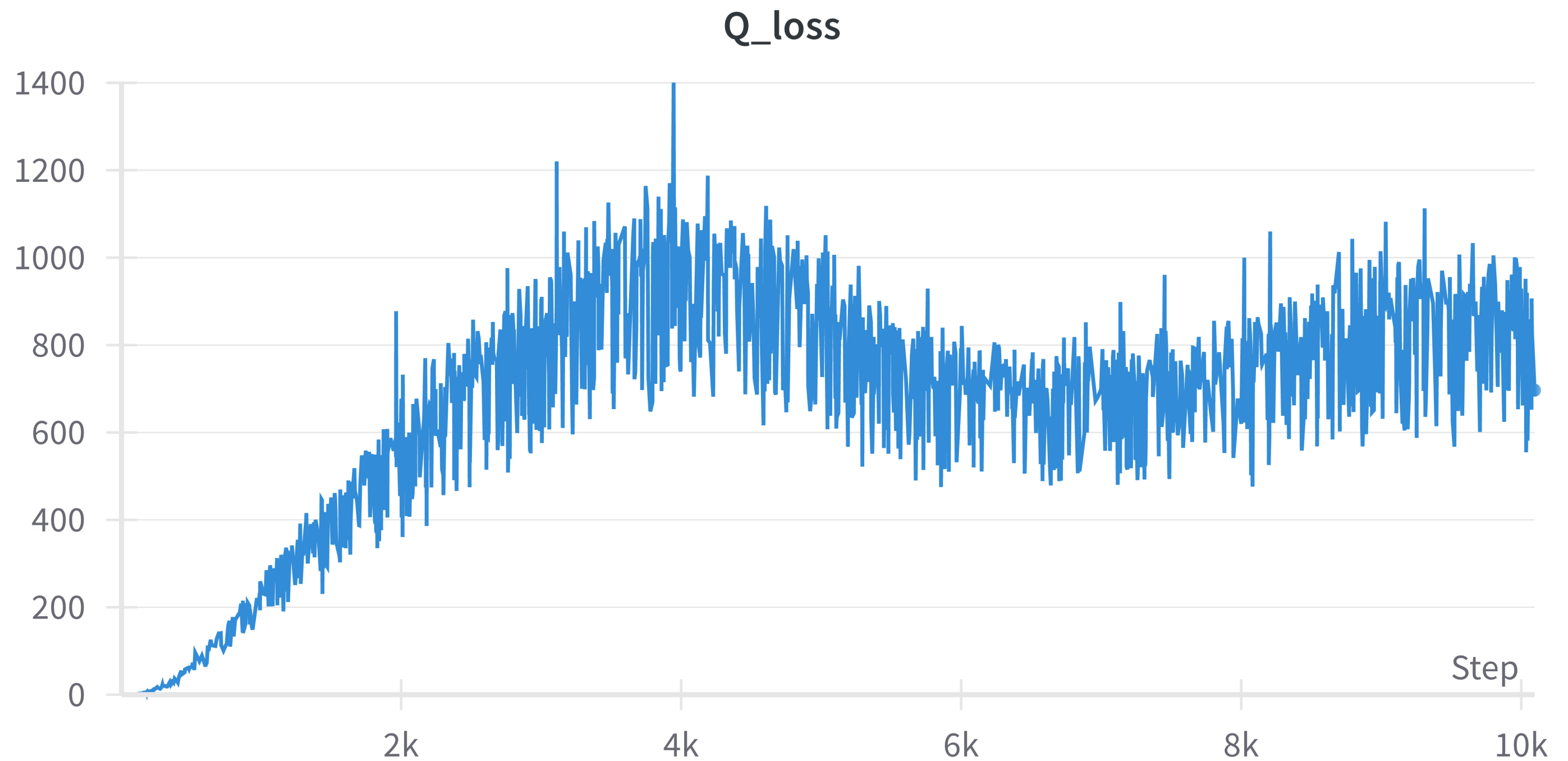
```

DDPG (3 points total)

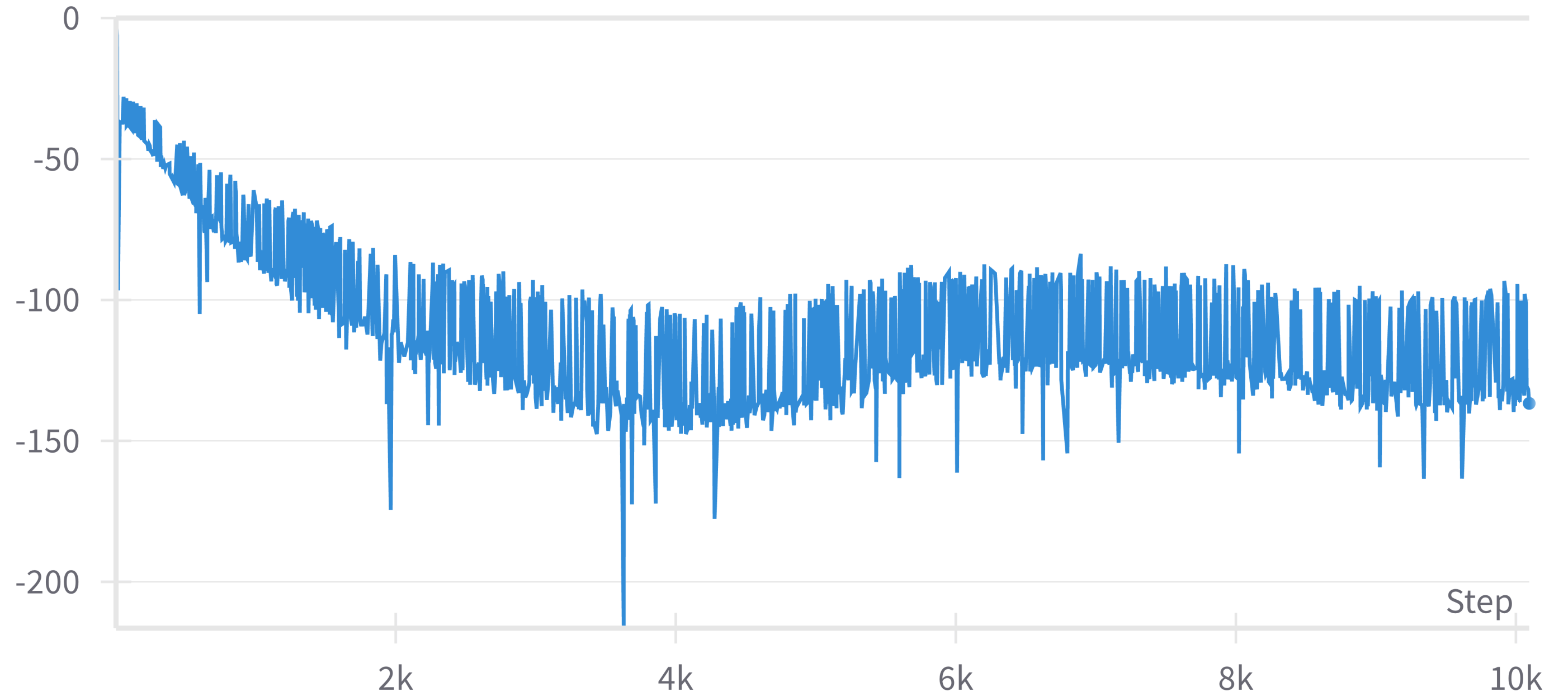
We will be implementing the DDPG algorithm. The DDPG algorithm is a model free, off policy algorithm that combines the actor-critic architecture with the insights of DQN. The algorithm is as follows: DDPG Fill in the TODOs in the `DDPG` class in `DDPG.py`

```
In [ ]: import DDPG
import utils
t = DDPG.DDPG(env,
              model.Actor,
              model.Critic,
              use_wandb=False,
              save_path = utils.get_save_path("DDPG", "./runs/"))

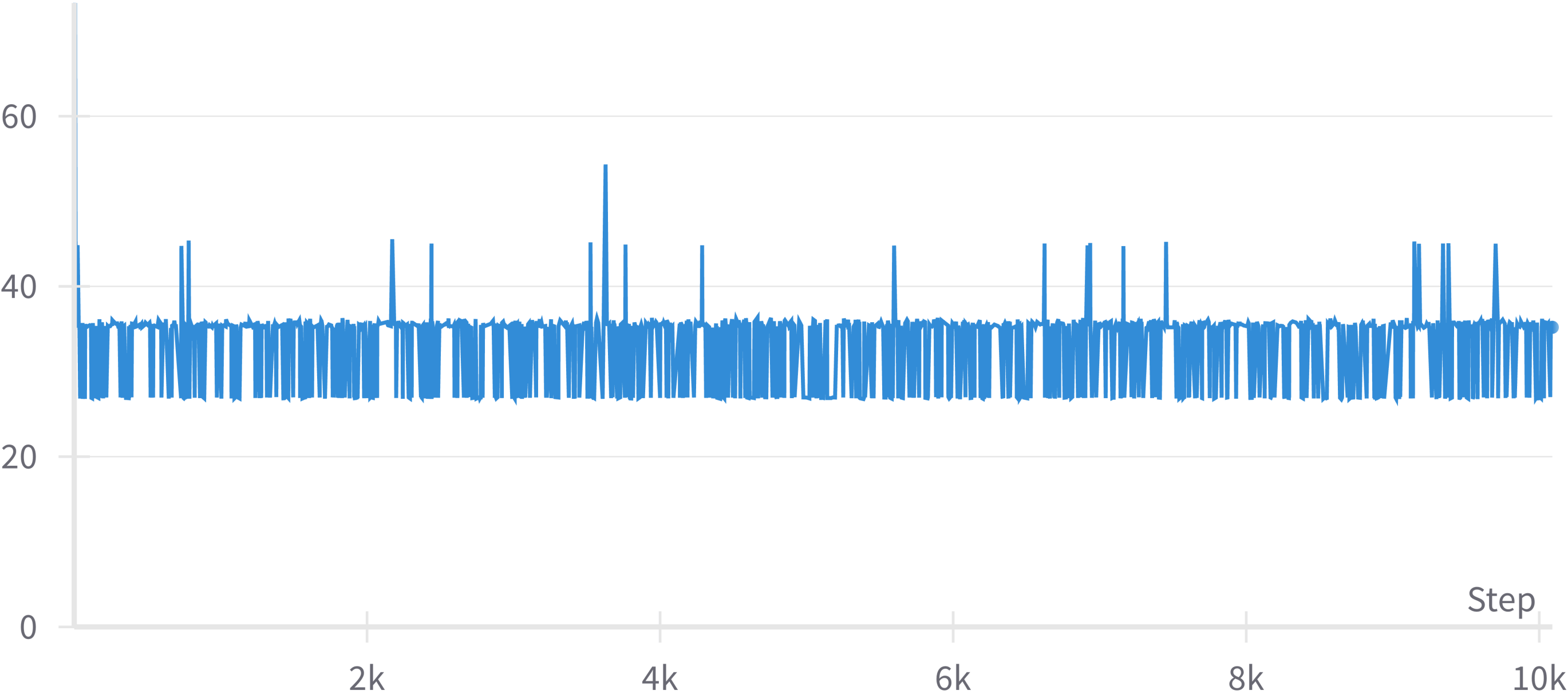
t.train(10000,
        100,
        100,
        1000,
        100,
        1)
```



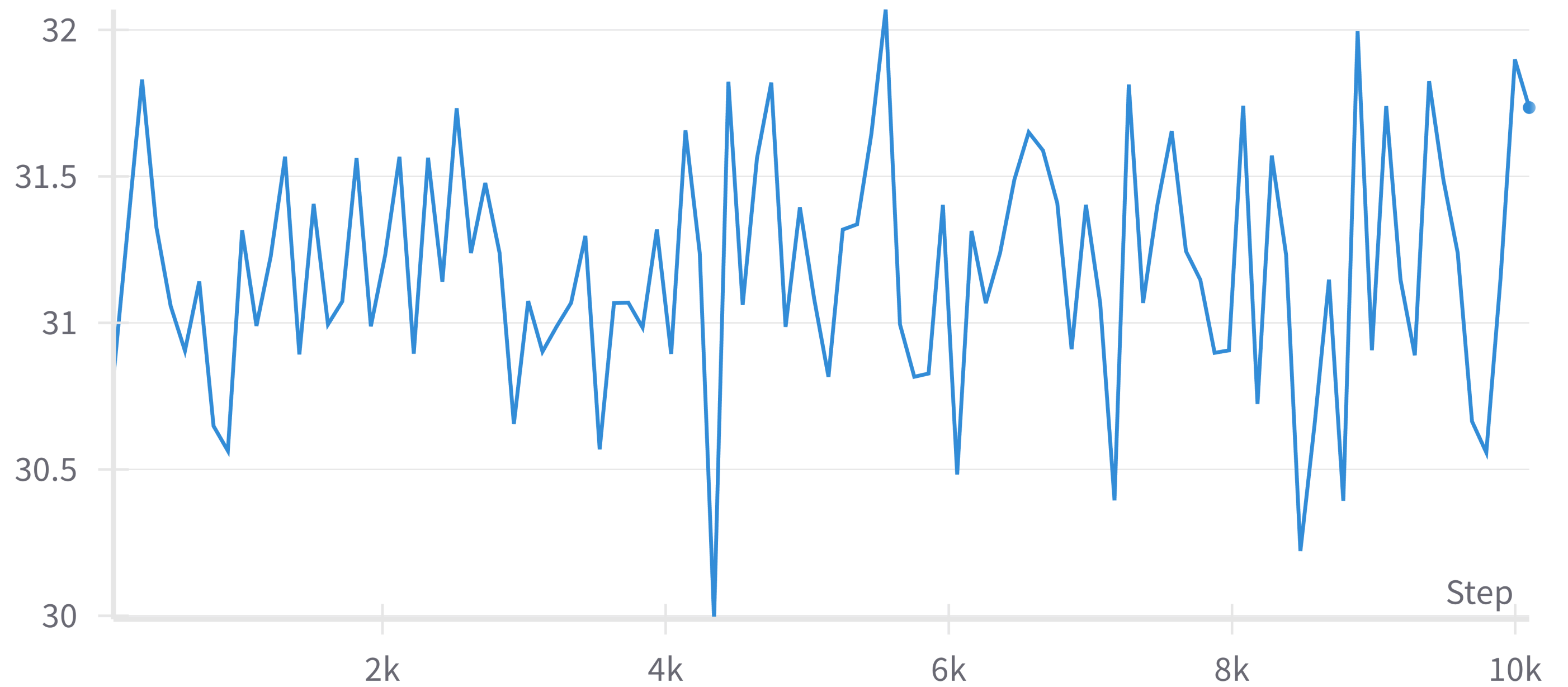
actor_loss

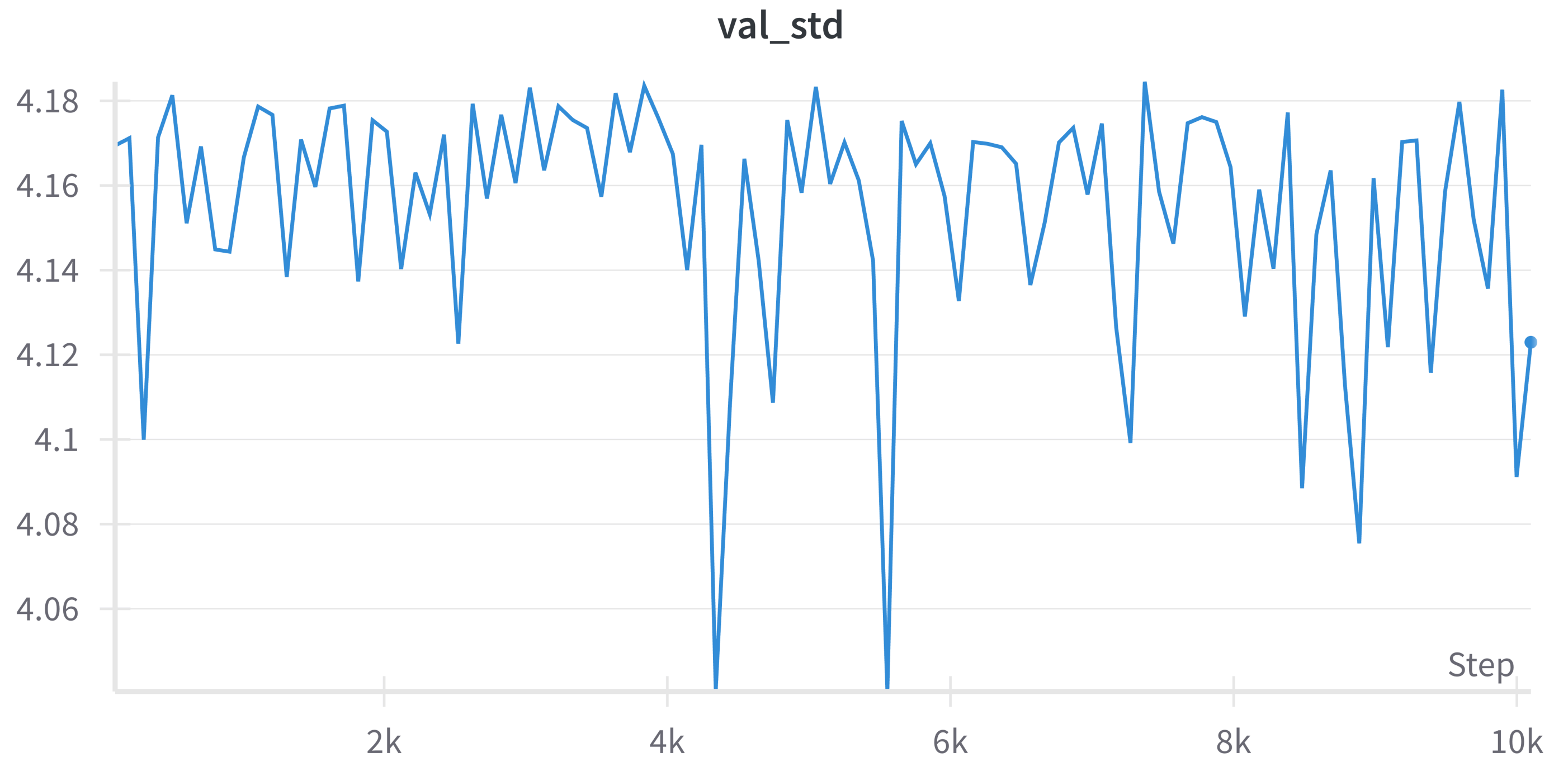


total_reward



val_mean





Results

Validation Mean: 31.734921155961793, Validation Std: 4.122968790294309

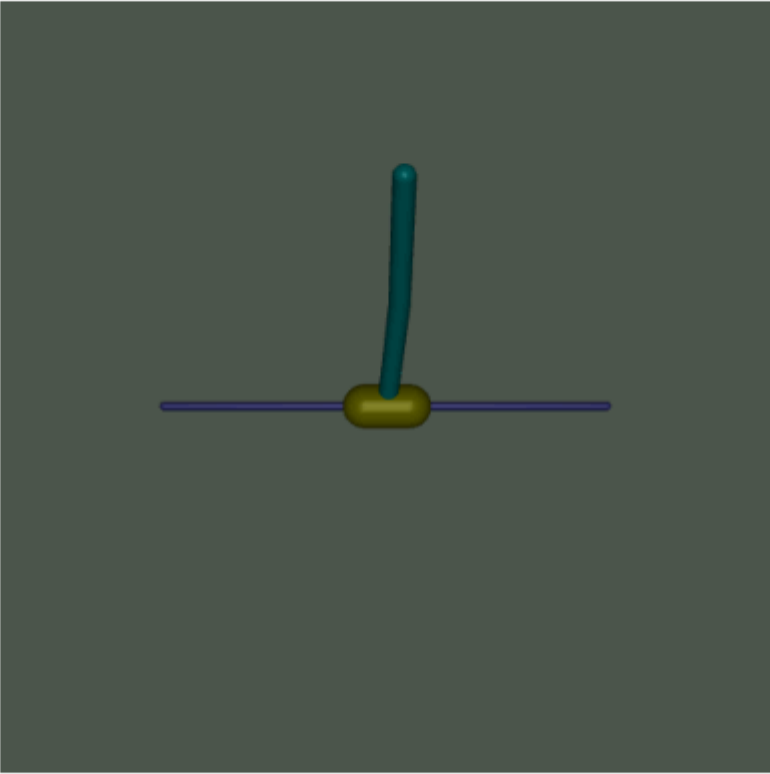
Test Mean: 30.956240341988522, Test Std: 4.172987532527436

Like what we did for the DQN, we can also animate one episode of the agent in the DoublePendulum environment.

```
In [ ]: total_rewards, frames = t.play_episode(0, True, 42, eval_env)
anim = animate(frames, max_frames = 1000)
print(total_rewards)
HTML(anim.to_jshtml())
```

35.206481058756225

Out[]:



-

⏮

⏪

⏩

⏭

⏴

⏵

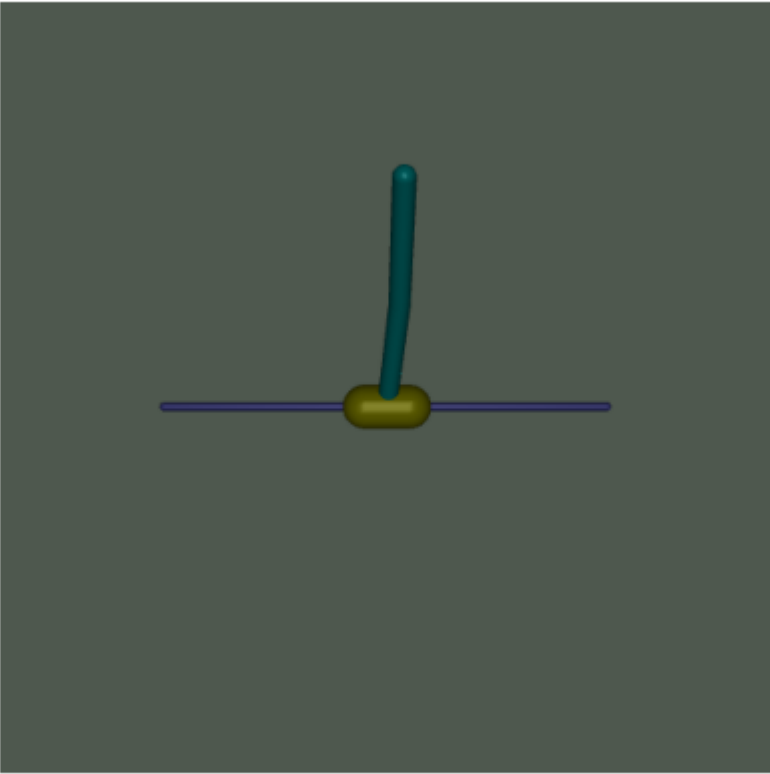
⏶

+

☐ Once

☒ Loop

☐ Reflect



As we can see, the agent is able to balance the double pendulum and it eventually reaches the equilibrium. However this equilibrium is not a stable equilibrium, so lets see how this model performs with perturbations. To do this, we will perturb the model every 49 steps with a large input of ± 0.75 N to the cart. We will see how the model performs with this perturbation.

```
In [ ]: import torch
frames = []
```

```
scores = 0
(s, _), done, ret = eval_env.reset(seed = 42
                                   ), False, 0

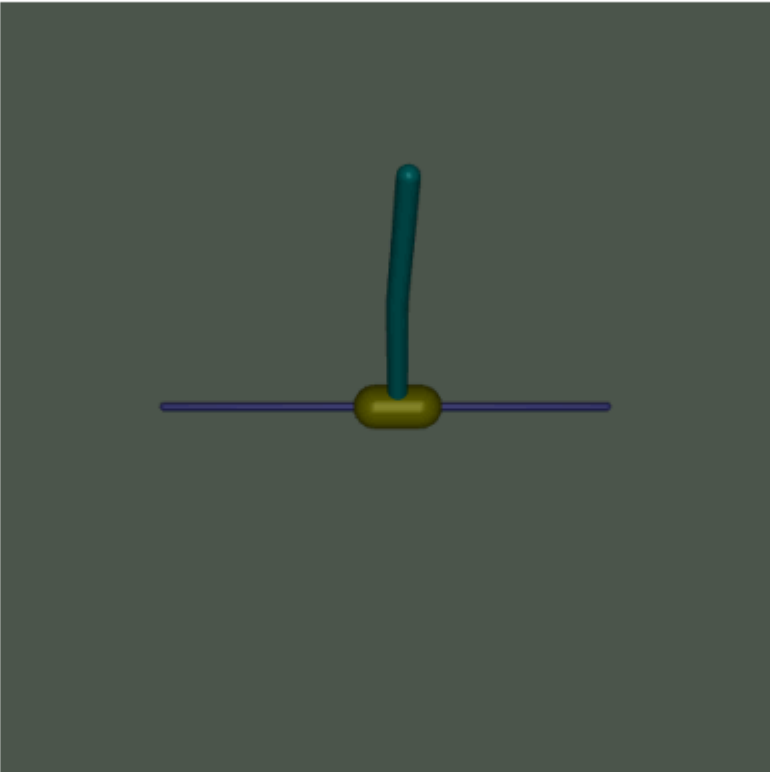
t.actor.eval()
S = []
outputs = []
# s, r, terminated, truncated, info = eval_env.step(3)
i = 0
with torch.no_grad():
    while not done:
        # if random.random() < 0.1:
        #     action = random.randint(0,4)
        # else:
        frames.append(eval_env.render())
        output = t.actor(torch.tensor(s).unsqueeze(0).to("cpu").float())
        i+=1
        if i%50 == 49:
            output += 0.75*(np.sign(torch.randn_like(output)))
        s_prime, r, terminated, truncated, info = eval_env.step(output.cpu().numpy().squeeze(0))
        s = s_prime
        ret += r
        done = terminated or truncated

scores += ret
```

```
In [ ]: anim = animate(frames,max_frames = 500)
print(total_rewards)
HTML(anim.to_jshtml())
```

35.206481058756225

Out[]:



-

⏮

⏪

◀

⏸

▶

⏩

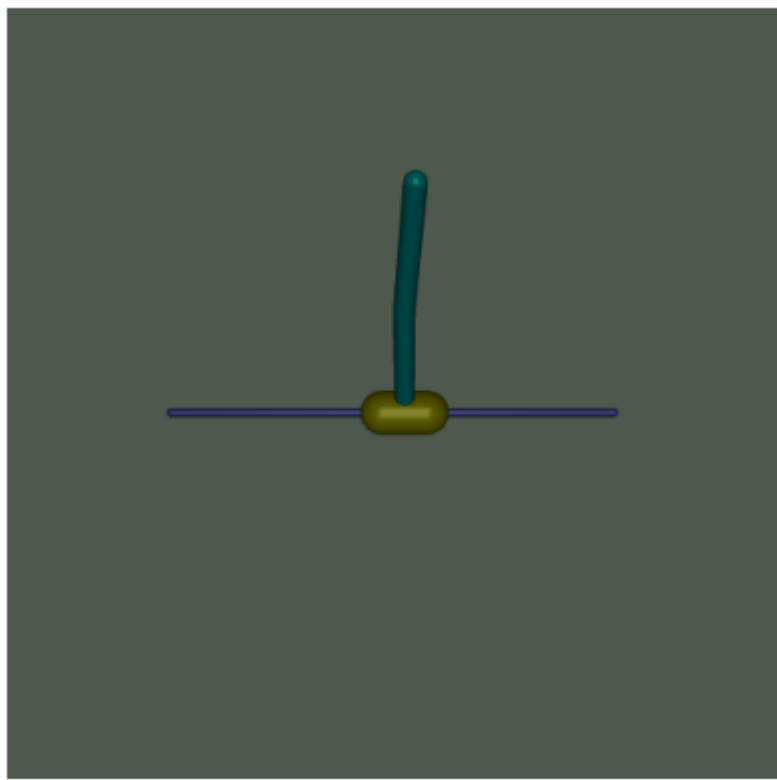
⏭

+

☐ Once

☒ Loop

☐ Reflect



You should see that the model is able to recover from the perturbation and is able to balance the double pendulum.